Tiedon eristäminen ja visualisointi lastensuojelun asiakaskertomuksista

Sami Kiviharju

Tampereen yliopisto
Informaatiotieteiden yksikkö/tietojenkäsittelyoppi
Pro gradu -tutkielma
Ohjaajat: Marko Junkkari ja Jaana Kekäläinen
17.6.2013

Tampereen yliopisto
Informaatiotieteiden yksikkö/tietojenkäsittelyoppi
Sami Kiviharju: Tiedon eristäminen ja visualisointi lastensuojelun asiakaskertomuksista
Pro gradu -tutkielma, 62 sivua
Kesäkuu 2013

Tiivistelmä

Tässä tutkielmassa sovelletaan kieliteknologiaa ja tiedon eristämistä lastensuojelun asiakaskertomuksista koostuvaan aineistoon. Tarkoituksena on tuottaa asiakaskertomuksista automaattisesti erilaisia visualisointeja, kuten aikajanoja ja graafeja, jotta tapauksista saataisiin nopeasti yleiskuva. Visualisointien tuottamista varten aineistossa esiintyvien henkilöiden nimien esiintymät poimitaan kirjauksista ja näiden perusteella päätellään automaattisesti tapauksen henkilöt ja heidän kokonimensä. Kokonimien osittaiset esiintymät liitetään tapauksen henkilöihin ja henkilöille pyritään tunnistamaan myös heidän roolinsa tapauksessa. Asiakaskertomusten arkaluonteisuudesta johtuen työssä käytetty aineisto anonymisoidaan automaattisesti poistamalla sieltä henkilötiedot sekä korvaamalla henkilöiden nimet johdonmukaisesti uusilla.

Avainsanat ja -sanonnat:

visualisointi, tiedon eristäminen, luonnollisen kielen käsittely

Sisällys

1 Johdanto

Lastensuojelu on osa sosiaalityötä, joka on velvoitettu dokumentoimaan toimintansa [Lastensuojelulaki 2007, § 33]. Lastensuojelulain 1. § mukaan lastensuojelun tarkoituksena on "turvata lapsen oikeus turvalliseen kasvuympäristöön, tasapainoiseen ja monipuoliseen kehitykseen sekä erityiseen suojeluun" [Lastensuojelulaki 2007]. Dokumentoinnin tulisi palvella sekä työntekijöitä että asiakkaita, mutta tietojärjestelmät eivät tue käytännön sosiaalityötä parhaalla mahdollisella tavalla. Ongelmana on etenkin riittävän yleiskuvan saaminen tapauksista. [Huuskonen ja Vakkari 2010] Yleiskuvan muodostamisen ongelmaa pyritään tässä työssä ratkaisemaan muun muassa kieliteknologiaa soveltamalla sekä erilaisilla visualisoinneilla. Samalla pyritään selkiyttämään ja havainnollistamaan jo tuotettuja asiakaskertomuksia, jotta niiden hyödyntäminen helpottuisi.

Lastensuojelua tarkastellaan tässä työssä sosiaalityöntekijöiden tuottamien lastensuojelun asiakaskertomusten kautta. Aineistona olevia asiakaskertomuksia muun muassa rakenteistetaan, anonymisoidaan, annotoidaan ja visualisoidaan. Anonymisoinnilla tarkoitetaan tässä työssä henkilötietojen, kuten nimien ja yhteystietojen korvaamista niiden anonyymeillä muodoilla. Annotoinnissa sanoihin liitetään lisätietoja, esimerkiksi erilaisiin ajanilmauksiin liitetään niiden aikaleima vakioidussa muodossa. Edellisten lisäksi työssä tunnistetaan tapauksissa esiintyvät henkilöt ja yhdistetään näiden tekstissä esiintyvät maininnat samaan viitekohteeseen, sekä esitetään Semantic MediaWiki:n [Krötzsch et al. 2007] avulla toteutettu hypertekstimuotoinen esitysmuoto aineistolle. Samalla työssä sovelletaan luonnollisen kielen käsittelyn menetelmiä ja tiedon eristämistä lastensuojelun asiakaskertomuksiin. Luonnollisen kielen käsittelyn kannalta kieltä lähestytään työssä dependenssijäsentämisen kautta [Tapanainen ja Järvinen 1997; Tarviainen 1977]. Tiedon eristämisessä (information extraction, IE) pyritään etsimään ja esittämään vapaassa tekstissä oleva oleellinen semanttinen informaatio rakenteisessa, mahdollisesti relationaalisessa muodossa [Wu ja Weld 2010]. Pyrkimyksenä on saada luonnollisella kielellä esitetty informaatio muotoon, jota on helpompi käsitellä koneellisesti. Tässä työssä tiedon eristäminen noudattaa perinteistä lähestymistapaa, jossa käyttäjä antaa järjestelmälle sen tarvitsemat säännöt ja hahmot valmiina [Etzioni et al. 2008].

Dawis-Mendelow luonnehtii sosiaalityötä informaatiointensiiviseksi työksi [Davis-Mendelow 1998]. Näin ollen onkin perusteltua pyrkiä helpottamaan ja tehostamaan työhön liittyvää tiedon käsittelyä. Työtehtäviin liittyvää tiedon kirjaamista ja käsittelyä ovat esitelleet etenkin Huuskonen ja Vakkari [2010, 2011]. Tässä työssä tuotettujen visualisointien käyttötapauksena on ajateltu olevan päivystävä sosiaalityöntekijä, joka käsittelee virka-ajan ulkopuolella akuuteiksi muuttuneet tapaukset ja lastensuojeluilmoitukset. Päivystävällä sosiaalityöntekijällä ei oleteta olevan aiempaa tietoa käsiteltäväksi tulevista tapauksista. Tämä käyttötapaus on helposti laajennettavissa kattamaan myös uudet työntekijät, jotka eivät tunne asiakkaita ennestään. Myös jo asiakkaan tun-

tevat sosiaalityöntekijät hyötyvät visualisoinneista, sillä asiakaskontaktin tiiviys vaihtelee eri tapausten välillä. Osa tapauksista kattaa vain lyhyen aikavälin, kun taas toiset kestävät koko lapsuuden ajan. Osalle tapauksista on myös tyypillistä olla ns. lepotilassa, ennen uutta ennakoimatonta aktivoitumista. [Huuskonen ja Vakkari 2010] Myös tällaisissa tapauksissa työntekijä hyötyy visualisoinneista, joiden avulla tapauksen muistiin palauttaminen helpottuu. Edellisen perusteella voidaan siis todeta lastensuojelun asiakaskertomusten visualisoinnille löytyvän laajaa sovellettavuutta läpi lastensuojelun eri työvaiheiden. Huuskosen ja Vakkarin jaottelun mukaan asiakastietojärjestelmään ja asiakaskertomukseen kohdistuu neljä erilaista tiedontarvetta [Huuskonen ja Vakkari 2010]. Nämä ovat yksittäisen faktan tarkistaminen, asiakkaan tilanteen holistinen tarkastelu, kirjausten silmäily ja koko asiakaskertomuksen tarkka läpikäyminen. Näistä työssä kuvatut visualisoinnit voisivat parhaiten tukea asiakkaan tilanteen holistista tarkastelua luomalla katsauksen asiakkuuden historiaan ja sosiaaliseen kontekstiin.

Kokonaisuudessaan tämä työ sijoittuu sosiaalityön, kieliteknologian, kielitieteen ja tietojenkäsittelyn leikkauspisteeseen. Työn voidaan edellisten lisäksi todeta kuuluvan lastensuojelun tietotekniikan alaan. Nguyen [2007] määrittee lastensuojelun tietotekniikan (child welfare informatics) alaksi, joka integroi lastensuojelun käytännön, informaatioteknologian ja tietojenkäsittelytieteen yhteen tukeakseen lastensuojelutyötä sen kaikilla tasoilla. Lastensuojelu ja sosiaalityö ovat varsin tuore sovellusalue kieliteknologialle verrattuna lääketieteeseen. Jo pelkästään Suomessa kieliteknologiaa on aiemmin sovellettu potilaskertomuksiin [Ginter et al. 2010]. Etenkin Turussa sitä on sovellettu laajasti tehohoidon potilaskertomuksiin [Hiissa et al. 2006; Laippala et al. 2009; Suominen 2007].

Työn toisessa luvussa esitellään käytetty aineisto sekä sen käsittelyyn liittyviä huomioon otettavia seikkoja, etenkin aineiston arkaluontoisuuteen liittyen. Kolmannessa luvussa esitellään aineiston käsittelyn vaiheet. Luvussa kuvattua käsittelyä voidaan työn kannalta luonnehtia esikäsittelyksi, joka pohjustaa viidennessä luvussa esiteltäviä visualisointeja ja muita työn varsinaisia tuloksia. Neljännessä luvussa esitellään aineiston lopullinen tallennusmuoto, johon päädytään kolmannessa luvussa kuvattujen työvaiheiden kautta. Kuudennessa luvussa kuvataan työtä varten toteutettuja ohjelmistoja, sekä esitellään työssä käytettyjä valmiita työkaluja ja kirjastoja. Seitsemännessä luvussa käydään läpi työssä kohdattuja haasteita ja mahdollisia kehitysehdotuksia, sekä sosiaalityöntekijöiden kommentteja visualisoinneista.

2 Aineisto

Käytetty aineisto koostuu erään kaupungin sosiaalitoimen lasten avohuollon asiakasdokumentaatiosta. Taustana dokumenttien tuottamiselle on lastensuojelulaissa [Lastensuojelulaki 2007, § 33] määrätty seuraavaa: "Lastensuojelun työntekijöiden tulee merkitä lasta tai nuorta koskeviin asiakasasiakirjoihin [...] kaikki lapsen tai nuoren tarvitsemien lastensuojelutoimenpiteiden järjestämiseen vaikuttavat tiedot sekä toimenpiteiden suunnittelun, toteuttamisen ja seurannan kannalta tarpeelliset tiedot". Taskisen [2010] mukaan kirjaamisessa on syytä kuvata lapsen ja vanhempien käyttäytymistä selväsanaisesti ja asianosaisten näkemyserot ja eri versiot tapauksista tulee kirjoittaa näkyviin. Havaintojen, tosiasioiden ja tilannearvioiden selkeä ja luotettava kirjaaminen on tarpeen myös lapsen, perheen sekä työntekijöiden oikeusturvan kannalta [Taskinen 2010]. Sosiaalityöntekijöiden lisäksi myös asianosaisilla, lähinnä lapsella ja perheellä, on oikeus tutustua asiakirjoihin. Myöhemmissä elämänvaiheissa lapsi ja nuori voi lisäksi rakentaa kuvaa omasta henkilöhistoriastaan asiakirjojen avulla. [Taskinen 2010]

Käytetyn aineiston pääasiallisia tuottajia ovat olleet sosiaalityöntekijät sekä eri sijoituspaikoissa työskentelevä henkilökunta. Pääosan aineistosta muodostavat asiakaskertomukset, päätökset ja asiakassuunnitelmat. Lukumääräisesti tapauksiin liittyviä päätöksiä on eniten, mutta listatuista dokumenttityypeistä ne ovat lyhimpiä. Taulukossa 2.1 ovat alkuperäisten asiakaskertomusten pituudet sivuina, sekä kirjausten ja lauseiden määrä. Asiakkaita ja näin ollen myös tapauksia on kymmenen.

Taulukko 2.1: Asiakaskertomusten pituudet sivuina, sekä kirjausten ja lauseiden lukumäärät.

Tapaus	Pituus sivuina	Kirjausten lukumäärää	Lauseiden lukumäärä
1	72	242	2163
2	165	671	4600
3	49	118	1485
4	57	170	1626
5	17	44	437
6	81	136	1755
7	16	51	449
8	124	322	3950
9	61	145	1716
10	113	277	3283

Asiakirjojen käsittelyssä on rajoituttu käsittelemään asiakaskertomuksia, koska päätökset ja asiakassuunnitelmat ovat huomattavasti asiakaskertomuksia lyhyempiä. Pituutensa ja vapaamuotoisuutensa vuoksi asiakaskertomuksia on myös tarpeellisempaa, se-

kä tutkimuksen kannalta kiinnostavampaa, saada helpommin sisäistettävään muotoon kuin muita dokumentteja.

2.1 Tutkimukseen liittyvät eettiset seikat

Kaikki tässä työssä esitetyt esimerkit perustuvat alkuperäiseen aineistoon. Esimerkeissä esiintyvät henkilöt on kuitenkin kaikki anonymisoitu tunnistamattomiksi työssä kuvatulla tavalla. Nimien lisäksi myös muut tapausten yksityiskohdat on muutettu niin, ettei niistä voida tunnistaa mitään henkilöihin, tapahtumaympäristöön tai tapausten yksityiskohtiin liittyviä tietoja. Kaikki aineistosta tehdyt yhteenvedot ja analyysit on suoritettu anonymisoidulla aineistolla.

Työssä käytetty Connexor Oy:n suomenkielen dependenssijäsennin on kaupallinen ohjelma, mutta se on lisensoitu tutkimuskäyttöä varten CSC - Tieteen tietotekniikan keskus Oy:n, lyhemmin CSC, palvelimelle. Tästä johtuen työssä suoritettu jäsentäminen on suoritettu tällä CSC:n palvelimella käytössä olevalla jäsentimellä. Koska jäsentäminen ei tapahdu paikallisesti vaan vieraalla palvelimella, on sen suorittamisessa otettu huomioon seuraavia tietosuojaan liittyviä seikkoja. Aineiston arkaluonteisuudesta johtuen jäsentimelle on syötetty vain anonymisoitua aineistoa. Jäsentäminen on tämän lisäksi suoritettu lause kerrallaan, siten että lauseet on lähetty palvelimelle jäsennettäväksi satunnaisessa järjestyksessä ja ne on koottu alkuperäiseen järjestykseen jäsentämisen jälkeen paikallisesti. Tätä varten lauseet on eroteltu paikallisesti omana työvaiheenaan kohdassa 3.3.3, vaikka jäsenninkin olisi voinut suorittaa lauseiden erottamisen toisistaan. Palvelimelle ei myöskään ole jäsentämisen aikana tallennettu väliaikaistiedostoja, jotka sisältäisivät osia aineistosta.

2.2 Asiakaskertomusten sisältö

Asiakaskertomukset jakautuvat osioihin, jotka taas jakautuvat kirjauksiin. Tapauksissa esiintyviä eri osioita ovat: avohuollon sosiaalityön kertomusosio, avotyön kertomusosio, jälkihuollon kertomusosio, perhekuntoutuksen kertomusosio, perhetukikeskuksen erityistyön kertomusosio, perhetukikeskuksen osastotyön kertomusosio ja sijaishuollon sosiaalityön kertomusosio. Näistä avohuollon sosiaalityön kertomusosio on tyypillisesti pisin ja mielenkiintoisin. Avohuollon osion kirjoittajana on tyypillisesti sosiaalityöntekijä ja osio kattaa usein koko tapauksen alusta loppuun asti, muiden osioiden muodostaessa lähinnä ohimeneviä episodeja. Muut osiot kuvaavat tyypillisesti varsin pikkutarkasti päivittäistä elämää sijoituspaikoissa, esimerkiksi perhetukikeskuksissa tai kuntouttavassa osastohoidossa. Näissä osioissa kirjoittajana on sijoituspaikan henkilökuntaan kuuluva henkilö. Osiot jakaantuvat kirjauksiin, jotka voidaan tässä työssä määritellä yhdellä kertaa kirjoitetuksi yhteenvedoksi esimerkiksi asiakastapaamisesta tai yhteydenotosta. Avohuollon kirjauksissa kirjaukset ovat rakenteeltaan säännöllisempiä ver-

rattuna muihin osioihin. Ne alkavat tyypillisesti päivämäärällä sekä kirjauksen tekijän nimellä. Muista osioista tämä tiukka rakenne pääsääntöisesti puuttuu.

Avohuollon sosiaalityön kertomusosion tyypillisiä kirjaustyyppejä ovat soitto tai muu yhteydenotto asiakkaaseen tai tämän perheeseen. Tämän lisäksi muita tyyppejä ovat asiakastapaamisista tehdyt muistiot ja muistiinpanot, sekä monialaiset eri ammattilaisten ja asiakkaiden välisten neuvottelujen raportointi. Asiakaskertomuksen alkuun on yleensä kerätty tärkeiden henkilöiden yhteystietoja, sekä kuvaus asiakkaan sosiaalisesta verkostosta. Yhteystietoja esiintyy kuitenkin varsin paljon myös pelkästään kirjausten lomassa. Kuvassa 2.1 on esitetty neljä esimerkkikirjausta, jotka kuvaavat edellä mainittuja kirjaustyyppejä. Mukana on lisäksi esimerkkejä, joissa asiakaskertomuksesta pelkästään viitataan muihin dokumentteihin ja niissä tallennettuun, tapauksen kannalta olennaiseen tietoon.

Kuva 2.1 Esimerkkejä anonymisoiduista kirjauksista

31061999 /**Eerika Kantola**
XXX lastenneuvolan terveydenhoitaja **Virva Hiltunen** soitti ja kertoi, että **Henna Feynmanin** vauva on kotiutunut sairaalasta tänään.

21081999 /**Eerika Kantola**
Huoltosuunnitelmaneuvottelu XXXX:ssa. Kirjattu erillinen huoltosuunnitelma

---

260899 /**Eerika Kantola**
Tehty palvelutilaus avotyöstä *XXXn*
Kantola Eerika

171100 / **Eerika Kantola**
Neuvottelu *XXX* perhekuntoutuksessa. Mukana **Henna** ja **Pilvi**, *XXXn* perhekuntoutusosastolta **Venla Koivisto** ja **Kerttu Simonen** sekä perheen avotyöntekijä **Aurora Sipiläinen**, *XXX* lastenneuvolan terveydenhoitaja **Virva Hiltunen** ja *XXX* sosiaaliasemalta sosiaalityöntekijä **Eerika Kantola**. Kerrattiin jo viime neuvottelussa esiin tulleita asioita tilanteesta jolloin **Henna** tuli perhekuntoutukseen. **Venla Koivisto** kertoi, että **Henna** oli käyttänyt aineita vielä sinä aamuna jolloin saapui perhekuntoutukseen. **Henna** oli jo täysin valmis itsekin vastaanottamaan hoitoa. Ensimmäisellä viikolla **Hennalla** oli selvästi vieroitusoireita Perhekuntoutuksen työntekijät totesivat, että **Henna** hyötyy kuntoutuksesta ja hän on ollut hyvin avoin asioissaan. **Venla Koivisto** kertoi, että **Henna** on valmis tekemään töitä ja haluaa elää **Pilvin** kanssa tavallista perhe-elämää. **Hennakin** havaitsi nyt mikä vaikutus hänen ainekäytöllään on ollut **Pilviin**. **Pilvin** hyvä kehitys vuorovaikutuksen alueella romahti **Hennan** aineiden käytön aikana. **Pilvi** oli *XXXn* tullessa apea ja väsynyt, nyt **Pilvi** on selvästi piristynyt ja hymyillytkin. **Henna** on nyt aloittanut mielialalääkityksen ja sanookin että jaksaa nyt paljon paremmin. Kuntoutuksen aikana **Hennan** seula on ollut kerran amfetamiinipositiivinen. Testituloksesta ei ole saatu vahvistusta vielä. Sovittiin, että **Henna** ja **Pilvi** jatkavat kuntoutuksessa vielä noin kaksi viikkoa ja kotiutuvat *1.12.2000*. Seulat jatkuvat edelleen kolme kertaa viikossa ja *10.12* alkaen mahdollisesti seulojen antamista harvennetaan. *XXXn* avotyö jatkuu, avotyöntekijänä jatkaa **Aurora Sipiläinen**.
Kantola Eerika

2.3 Aineiston erityispiirteet ja käsittelyn ongelmat

Aineisto on vapaata tekstiä, joten sen käsittelyn ongelmaksi nousee soveltuvien työkalujen suppea saatavuus. Erityisesti suomen kieltä varten tarkoitettuja työkaluja on tarjolla vain vähän. Ongelmaksi niiden kohdalla muodostuu hinta ja saatavuus. [Haverinen, Ginter, Laippala ja Salakoski 2009] Luvussa 6.4 on tarkasteltu kieliteknologisten työkalujen ja resurssien tilaa ja saatavuutta.

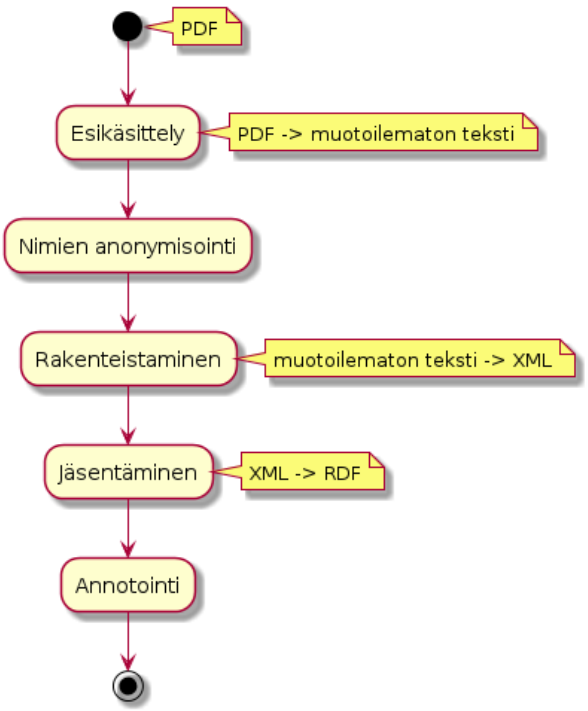
Kirjaukset eivät yleiseltä kieliasultaan ole erityisen huoliteltuja. Niitä on myös oletettavasti osittain kirjoitettu asiakaskastapaamisten aikana, jolloin kielenlaatu ja kirjoitusasu ovat saattaneet kärsiä. Tämä näkyy paikoin ilmenevinä epätavallisina lauserakenteina sekä kirjoitusvirheinä. Kirjauksissa käytetty kieli sisältää lisäksi jonkin verran omia ammatillisia lyhenteitä. Lyhenteiden käyttö ei ole täysin vakiintunutta ja esimerkiksi sanalle sosiaalityöntekijä löytyy viisi erilaista lyhennettyä muotoa. Osa muodoista eroaa toisistaan tosin vain välimerkkien käytön osalta. Mainittujen lyhenteiden ymmärtäminen tekstistä on toki ihmiselle helppoa, mutta muodostaa haasteen tekstin automaattiselle käsittelylle.

Kirjauksia ei aineiston luovutetussa PDF-muodossa ole koneluettavuuden kannalta eroteltu yksikäsitteisesti, vaan eri kirjaukset on eroteltu pelkillä tyhjillä riveillä toisistaan. Kappaleet on kirjausten sisällä eroteltu samalla tavoin kappalejaoin, joten kirjausten erottelu on jouduttu toteuttamaan omana työvaiheenaan. Rajat ovat ihmissilmälle ilmeisiä ja ne saadaan koneellisestikin varsin hyvin eroteltua toisistaan. Samanlaisiin ongelmiin törmättiin myös kirjausten metadatan tunnistamisessa. Vaikka kirjaukset on alunperin syötetty asiakastietojärjestelmään ja ne ovat koneen käsiteltävässä muodossa, ovat kirjauksia tehneet työntekijät kiertäneet järjestelmän puutteita kirjaamalla metatiedot, päivämäärän ja kirjaajan, itse kirjauksen alkuun. Vaihtelevat tavat ilmaista näitä metatietoja on täytynyt ottaa huomioon, kun kirjausten metadataa on automaattisesti rekonstruoitu. Kirjausten erottelu ja niiden metadatan tunnistaminen onnistui miltei kokonaan automaattisesti, vaatien käyttäjän väliintuloa vain murto-osassa kirjauksia.

3 Aineiston käsittelyn vaiheet

Koska aineisto on ensisijaisesti sosiaalityöntekijän ja asiakkaan käyttöön tarkoitettua proosatekstiä, liittyy sen automatisoituun jatkokäyttöön useita ongelmia. Alkuperäinen PDF-muotoinen aineisto muunnetaan usean eri työvaiheen sekä tiedostomuodon kautta tekstimuotoon, jotta siihen sisältyvä informaatio saataisiin paremmin koneellisesti käsiteltävään muotoon. Eri vaiheiden aikana aineistoa muun muassa korjataan, laajennetaan, anonymisoidaan ja annotoidaan. Kuvassa 3.1 on esitetty aineiston käsittelyn vaiheet karkealla tasolla. Vaiheet kuvataan tässä yleisesti ja myöhemmin yksityiskohtaisesti. Kuvassa on lisäksi esitetty aineiston tallennusmuodon muutokset.

Kuva 3.1 Aineiston käsittelyn vaiheet.



Esikäsittelyssä aineistolla suoritetaan useita erilaisia korjauksia ja korvauksia. Esimerkiksi lyhenteitä laajennetaan ja samalla aineistosta poistetaan henkilö- ja yhteystietoja. Anonymisoinnissa aineiston henkilöiden nimet korvataan johdonmukaisesti uusilla nimillä.

Rakenteistamisvaiheessa aineisto muutetaan vapaasta tekstistä puolirakenteiseen Extensible Markup Language (XML) -muotoon [Elmasri ja Navathe 2004]. Tämä tapahtuu pilkkomalla aineisto kirjauksiin, jotka pilkotaan edelleen lauseiksi. Kirjaukset ja lauseet esitetään lopulta hierarkkisesti XML-muodossa.

Jäsentämisessä sanoille päätellään muun muassa sanaluokka sekä mahdollinen taivutusmuoto. Näiden lisäksi sanojen roolit ja keskinäiset riippuvuudet selvitetään. Jäsen-

täminen suoritetaan tässä työssä dependenssikielioppia noudattaen [Tarviainen 1977]. Jäsentämistä ja dependenssikielioppia on kuvattu tarkemmin osiossa 3.4. Koska jäsennetyt lauseet ja niiden toisistaan riippuvat sanat on luontevaa esittää graafina, tallennetaan jäsennetyt lauseet RDF-graafeina. Graafimallissa sanoista ja niihin liittyvästä informaatiosta tehdään erillisiä solmuja, jotka liitetään nimetyillä kaarilla toisiinsa. RDF-kieltä ja sillä toteutettua aineiston tallennusmuotoa esitellään osiossa 4.

Annotointivaiheessa osaan sanoista liitetään lisäinformaatiota. Päivämääriä, lukuja, kellonaikoja ja nimiä edustavat sanat merkitään niiden edustaman luokan mukaisesti. Tätä lisäinformaatiota käytetään myöhemmissä vaiheissa uuden tiedon johtamiseen sekä kohinaa lisäävien sanojen poissulkemiseen. Kohinalla tarkoitetaan tässä kohtaa sanoja, jotka vaikeuttavat myöhempien vaiheiden analyysejä niiden kannalta merkityksettömällä vaihtelulla. Esimerkkeinä tästä ovat vaihtuvat päivämäärät ja nimet, joiden osalta vaihtelu tapausten välillä on suurta. Sanastollisesti muun tekstin voidaan olettaa käsittelevän eri tapauksissa samoina toistuvia käsitteitä, sanoja ja fraaseja. Annotoinnin aikana päätellään lisäksi annotoitujen nimisanojen perusteella tapauksen henkilöt ja näiden koko nimet. Lisäksi tekstin nimisanat yhdistetään siihen henkilöön, jota ne edustavat.

3.1 Esikäsittely

Myöhempien työvaiheiden helpottamiseksi aineistoa esikäsitellään. Esikäsittelyn aikana aineistoa muunnetaan ja korjataan helpommin käsiteltävään muotoon rakenteen sekä varsinaisen sisällön osalta. Rakenteeseen liittyvät muunnokset poistavat tekstiksi muunnetusta aineistosta sivunvaihtoja, sekä korjaavat automaattisen tavutuksen luomia virheitä. Sisältöön liittyviä muunnoksia ovat esimerkiksi lyhenteiden laajentaminen ja lyöntivirheiden korjaaminen.

3.1.1 Aineiston muuntaminen PDF-muodosta tekstimuotoon

Jatkokäsittelyn mahdollistamiseksi aineisto muutetaan **pdftotext**-ohjelman avulla PDF-muodosta tekstimuotoon. Ohjelma muuntaa PDF-dokumentit muotoilemattomaksi tekstiksi, säilyttäen sivu- ja kappalejaot.

PDF-muotoisen aineiston sekä sen tekstiversion jokaisen sivun lopussa on määrämuotoinen päivämäärä sekä sivunumero. Tämä on kuitenkin jatkokäsittelyn kannalta merkityksetöntä informaatiota, sillä päivämäärä kertoo vain ajan, jolloin tapausten tulosteet on tuotettu. Sivunumerot ja päivämäärät poistetaan lyhyen säännöllisen lausekkeen avulla.

3.1.2 Lyhenteiden laajentaminen ja sekalaisten korjausten tekeminen

Aineisto sisältää jonkin verran lyhenteitä, joista yleisimpiä laajennetaan jäsennyksen helpottamiseksi. Laajennettavia lyhenteitä valitessa on lisäksi otettu huomioon vain sellaiset lyhenteet, jotka voidaan korvata niin, että niiden laajennettu sijamuoto on mahdollisimman suurella todennäköisyydellä oikein. Esimerkiksi lyhenne *vl* voitaisiin laajentaa sanoiksi *viikonloppu* tai lauseyhteydestä riippuen myös muuhun kuin perusmuotoon. Lyhenteen *v-loppuisin* kohdalla taas ei ole ongelmaa monitulkintaisuuden suhteen, vaan se voidaan huoletta laajentaa muotoon *viikonloppuisin*. Taulukossa 3.1 on listattu osa laajennetuista lyhenteistä sekä niiden laajennetut muodot. Kaiken kaikkiaan laajennettavia lyhenteitä on 59 kappaletta.

Aineistoa ja taulukkoa 3.1 silmäilemällä voidaan havaita, että lyhenteiden käyttö on varsin kirjavaa eri kirjaajien välillä. Toiset käyttävät enemmän lyhenteitä kuin toiset ja yleisesti tunnettujen lyhenteiden lisäksi kirjaajat käyttävät omia ammatillisia lyhenteitä. Lisäksi havaitaan, ettei esimerkiksi sanalle sosiaalityöntekijä löydy yhtä johdonmukaisesti käytettyä lyhennettä.

Taulukko 3.1: Muutamia laajennettavia lyhenteitä.

lyhenne	laajennettu muoto
Aktissa	Asiakastietojärjestelmässä
sos.tt.	sosiaalityöntekijä
sos.tt	sosiaalityöntekijä
stt.	sosiaalityöntekijä
stt	sosiaalityöntekijä
sos.työntekijä	sosiaalityöntekijä
Petukkeen	perhetukikeskuksen
last.psyk.	lasten psykiatri
nuorisopsyk. työryhmään	nuorisopsykiatriseen työryhmään
v-loppuisin	viikonloppuisin
hlökuntaa	henkilökuntaa
krt	kertaa
mm.	muun muassa
e.	euroa.

Lyhenteiden lisäksi aineistosta korjataan lyöntivirheitä, kuten puuttuvia sanavälejä. Osaan tapauksista on myös käsin lisätty kirjausrajoja ja tehty pieniä tapauskohtaisia korjauksia. Tapauskohtaisina korjauksina on korjattu esimerkiksi nimilyhenteitä, kuten esimerkiksi lyhenne *N-P*, joka on korvattu Niko-Petterillä.

3.1.3 Henkilö- ja yhteystietojen poistaminen

Anonymisointi alkaa kaikkien dokumentin sanojen läpikäynnillä, jossa tarkastellaan, ovatko ne henkilötunnuksia tai muita poistettavia merkkijonoja. Tämän jälkeen poistettavaksi tunnistetut merkkijonot korvataan niiden anonymisoiduilla versioilla. Käytännössä henkilö- ja yhteystietojen poistaminen tapahtuu säännöllisillä lausekkeilla, joilla tunnistetaan muun muassa:

- puhelinnumeroita,
- henkilötunnuksia,
- osoitteita.

Nykyisellään esimerkiksi henkilötunnuksista poistetaan kaikki muu paitsi tieto syntymävuodesta. Syntymävuosi säästetään, sillä tämä on lastensuojeluaineiston ymmärtämisen kannalta oleellista. Osoitteet korvataan sanalla *Kuusitie* ja puhelinnumerojen jokainen numero korvataan merkillä *0*.

3.1.4 Eri riveille pilkottujen päivämäärien korjaaminen

Aineiston PDF-muotoinen versio on rivitetty sanavälien ja välimerkkien kohdalta. Tästä aiheutuu ongelmia etenkin silloin, kun rivinvaihto on lisätty keskelle päivämäärää. Esimerkiksi 30.5.2012 voitaisiin edellä mainitulla tavalla rivittää neljällä eri tavalla, joista pisteen kohdalla rivittäminen haittaa päivämäärän tulkintaa. Tästä johtuen eri riveille pilkotut päivämäärät pyritään korjaamaan poistamalla niiden sisäinen rivitys säännöllisen lausekkeen avulla. Kuvassa 3.2 oleva esimerkki havainnollistaa suoritettavaa korvausta:

Kuva 3.2 Esimerkki rivityksen korjaamisesta

```
Äiti soitti 12.8.
2005.
Korjataan muotoon:
Äiti soitti 12.8.2005.
```

3.2 Nimien anonymisointi

Kun aineistoa on edeltävien vaiheiden aikana siistitty, suoritetaan nimien anonymisointi. Anonymisoinnissa nimet korvataan johdonmukaisesti uusilla nimillä. Johdonmukaisessa korvaamisessa miesten nimet korvataan miesten nimillä ja naisten nimet naisten nimillä. Korvaava nimi säilyy myös samana koko tapauksen ajan. Uusi nimi kuitenkin taivutetaan aina samaan sijamuotoon, kuin missä alkuperäinen nimi kulloisessakin korvattavassa kohdassa on.

Kokonaisuutena nimien anonymisointi on kolmivaiheinen prosessi, johon kuuluvat:

- nimien tunnistaminen,
- korvaussääntöjen tuottaminen ja
- korvaussääntöjen soveltaminen.

Korvaussäännöt ilmaisevat yksinkertaisen merkkijonokorvauksen, jota käyttäen anonymisointi varsinaisesti suoritetaan.

3.2.1 Nimien tunnistaminen

Nimien tunnistamiseen käytetään kahta tapaa. Ensimmäinen on yksinkertainen sanalista nimistä ja niiden taivutuksista. Nimilistan lähteitä on esitelty tarkemmin kohdassa 6.1.2. Listalla olevien nimien taivutetut muodot on tuotettu käyttäen SWERG+ -taivutusmuotogeneraattoria [Kettunen ja Arvola 2012]. Taivutusmuotogeneraattoria kuvaillaan tarkemmin korvaussääntöjen tuottamisen yhteydessä kohdassa 3.2.2. Toinen tapa nimien tunnistamiseen on käyttää apuna Suomi-malaga -työkalua [Väisänen ja Pitkänen 2006], joka sisältää suomenkielen morfologisen kuvauksen Malaga-kielellä. Malaga [Beutel 1995] on ohjelmointikieli luonnollisen kielen kielioppien toteuttamista varten ja se pohjautuu Roland Hausserin Left-Associative Grammar -formalismiin [Hausser 1992]. Suomi-malagaa voi käyttää muun muassa sanojen sanamuotojen ja sanaluokkien tunnistamiseen, jolla tavoin sitä tässä työssäkin käytetään. Käytännössä sanan tunnistamiseen käytetään ensin Suomi-malagaa ja mikäli se ei tunnista sanaa nimeksi, verrataan sanaa tämän jälkeen edellä mainittuun nimistä koostuvaan sanalistaan. Näin toimitaan, koska Suomi-malagan tuntema nimistö on rajallinen ja sanalistan avulla saadaan tunnistettua myös aineistossa esiintyviä uusia tai vierasperäisiä nimiä. Aineistossa esiintyy myös eräiden vierasperäisten nimien taivuttamisessa varsin suurta vaihtelua, jota on korjattu lisäämällä tekstissä esiintyvät taivutusmuodot sanalistaan. Nimiksi tunnistettujen sanojen joukkoa suodatetaan, jotta saadaan poistettua väärin tunnistettuja nimiä kuten *Aamu* tai *Aina*. Käytännössä tämä tapahtuu poistamalla edellä mainittujen nimien kaltaiset nimet, jos ne esiintyvät dokumentissa vain lauseiden alussa, jolloin voidaan päätellä sanojen olevan muodollisesti, mutta ei tässä yhteydessä semanttisesti nimiä. Nimientunnistin osaa myös tunnistaa virheellisesti pienellä alkukirjaimella kirjoitetut nimet ja yhdistää ne niiden oikeinkirjoitettuun muotoon.

3.2.2 Korvaussääntöjen tuottaminen

Kun nimet on tunnistettu tekstistä, korvataan ne johdonmukaisesti uusilla, satunnaisesti arvotuilla nimillä. Nimen perusmuoto ja sen eri taivutusmuodot otetaan tässä yhteydessä huomioon aiemmin todetulla tavalla. Taivutetut nimet lemmataan ja näin saadaan nimen perusmuoto, jota käytetään uuden taivutetun nimen tuottamisessa. Ongelmia syntyy esimerkiksi sanan *Mikon* kohdalla. Sanan perusmuotoa ei voida yksikä-

sitteisesti määritellä, sillä esimerkiksi nimet *Mikko* ja *Miko* taipuvat samalla tavoin. Näissä tilanteissa valitaan perusmuodoista se, joka esiintyy jo dokumentissa. Jos dokumentissa ei esiinny sanaa *Mikko*, valitaan näin ollen perusmuodoksi *Miko*.

Alkuperäisten nimien lemmat korvataan uusilla, jonka jälkeen uusia lemmoja käytetään taivutusmuotogeneraattorin syötteenä. Taivutusmuotogeneraattori tuottaa lemmatulle nimelle joukon erilaisia taivutusmuotoja, jonka jälkeen korvaavan nimen taivutusmuodot valitaan taivutusmuotogeneraattorin tuottamasta taivutusmuotolistasta. Tässä työssä taivutusmuotogeneraattorina käytetään SWERG+ -ohjelmistoa [Kettunen ja Arvola 2012]. SWERG+ tuottaa taivutusmuotoja oppimiensa sääntöjen mukaisesti ja sen tuottamat taivutusmuodot sisältävät useita vääriä taivutuksia [Kettunen ja Arvola 2012]. Taulukossa 3.2 nähdään nimelle *Alexandra* tuotetut taivutetut muodot. Kuten taulukosta huomataan, eivät kaikki tuotetut muodot ole täysin oikeita. Nykymuodossaan SWERG+ ei myöskään tuota pelkästään yhtä pyydettyä taivutusmuotoa, vaan kaikki, jotka se pystyy muodostamaan. Se ei myöskään ilmaise, mitä taivutusmuotoa tuotettu taivutettu sana edustaa, vaan palauttaa pelkästään listan taivutettuja sanoja.

Taulukko 3.2: SWERG+:n tuottamat taivutukset sanalle *Alexandra*

Alexandra	Alexandrojen	Alexandroja	Alexandroissa	Alexandroina
Alexandroiksi	Alexandroilla	Alexandroilta	Alexandroille	Alexandroitta
Alexandroista	Alexandroihin	Alexandrat	Alexandran	Alexandraa
Alexandrassa	Alexandrana	Alexandraksi	Alexandralla	Alexandralta
Alexandralle	Alexandratta	Alexandrasta	Alexandraan	Alexandrt
Alexandrl	Alexandrn			

Korvaavaa taivutettua nimeä valittaessa käydään läpi kaikki tuotetut muodot ja niitä verrataan Suomi-malagan avulla alkuperäiseen taivutusmuotoon. Suomi-malagaa käytetään tunnistamaan uuden sekä alkuperäisen sanan sanamuoto. Mikäli taivutusmuodot Suomi-malagan mukaan täsmäävät, valitaan korvaavaksi sanaksi se, jota parhaillaan verrataan alkuperäiseen. Suomi-malaga ei kuitenkaan tunne tai tunnista kaikkia nimiä. Mikäli näin käy koetetaan oikea korvaava taivutus päätellä käymällä läpi taivutusmuotokandidaatit ja pisteyttämällä ne. Pisteyttäminen tapahtuu vertaamalla kandidaattia alkuperäiseen sanaan siten, että sanojen merkkejä verrataan lopusta alkuun ja merkkien täsmätessä kasvatetaan kandidaatin pisteytystä. Suurimman pistemäärän saa siis sana, jonka loppuosa muistuttaa eniten alkuperäistä sanaa.

3.2.3 Korvaussääntöjen soveltaminen

Edellä kuvatulla tavalla tuotetuista vanhojen ja uusien nimien pareista muodostetaan korvaussääntöjä. Korvaussääntöjen soveltaminen aineistoon tehdään säännöllisiä lausekkeita käyttäen. Sekä nimien tunnistamisessa että korvaamisessa otetaan huomioon tavanomaisista poikkeavat nimi-ilmaukset, kuten *Kylli*-täti tai *Matti*. Esimerkin sanat siis tunnistetaan ja korvataan säilyttäen nimien ympäristö ennallaan. Nykyinen toteutus säilyttää yhdysnimet yhdysniminä, esimerkkinä *Niko-Petteri*, mutta korvaa kuitenkin molemmat nimen osat.

Lopulta käyttäjälle palautetaan anonymisoitu teksti sekä tehtyjä nimimuutoksia kuvaavat korvaussäännöt. Nimet, joita ei voitu anonymisoida, palautetaan myös käyttäjälle. Näiden joukkoon kuuluvat muun muassa sanat, joille ei automaattisesti voitu päätellä oikeaa taivutusta. Niiden korvaussäännöt voidaan kuitenkin käsin täydentää ja soveltaa aineistoon samalla muiden sääntöjen kanssa.

Kuva 3.3 Esimerkki anonymisoidusta tekstistä

020292 / **Ainoliisa Sutinen**
Tapaaminen **Lassi Ojalan** kanssa *XXX* sosiaaliasemalla. Tarkoitus oli keskustella **Turkan** hoidosta ja sen vaatimista lastensuojelun päätöksistä. **Lassi** kertoi **Turkalla** menevän ihan hyvin *XXX* . **Turkka** oli ollut kotilomilla viikonloppuna ja kaikki oli mennyt hyvin. **Lassi** oli tyytyväinen **Turkan** hoitoon *XXX* . Perhetyöntekijät ovat tulossa heille kotiin keskiviikkona 14.2. ensimmäistä kertaa. **Turkka** on **Lassin** mukaan rauhoittunut. **Lassia** harmitti hieman se,että **Turkka** on laitoksessa hoidossa ja hän sanoi kyllä pärjäävänsä **Turkan** kanssa. Keskustelimme avohuollon sijoituksesta ja huostaanotosta ja käytännön asioista. Kerroin, että **Lassille** ei asiakasmaksua määrätä, mutta elatusapu tullaan perimään hoidon korvauksena.
Sutinen Ainoliisa

Kuvassa 3.3 on esimerkki anonymisoidusta tekstistä. Ohjelman korvaamat nimet on merkitty lihavoimalla. Päivämäärät ja paikat on tätä esimerkkiä varten muutettu käsin.

3.3 Rakenteistaminen

Tähän asti aineiston tapauksia on käsitelty yhtenäisenä jonona sanoja. Rakenteistamisvaiheessa tapaukset jaotellaan ensin kirjauksiin ja myöhemmin lauseisiin. Lausetasolle etenevä rakenteistaminen mahdollistaa myöhemmän, lause kerrallaan tapahtuvan luonnollisen kielen käsittelyn ja jäsentämisen.

XML-kielen tietomalli kuvaa puurakenteen, joka koostuu elementeistä ja näiden attribuuteista [Elmasri ja Navathe 2004]. Elementit voivat sisältää lomittain tekstiä ja toi-

sia elementtejä. Elementtien sisältyminen toisiin elementteihin luo XML-dokumentille hierarkkisen puurakenteen. Attribuutit liittyvät elementteihin ja ne voivat sisältää vain tekstiarvoja. XML-dokumentin rakenne voidaan kuvata XML DTD (document type declaration) -määrittelyn avulla. Määrittely kuvaa dokumentin hierarkkisen rakenteen, kuten elementtien esiintymisrajoitukset. Lisäksi se kuvaa elementtien pakolliset ja valinnaiset attribuutit. Tarkemman kuvauksen XML-kielestä ja DTD-määrittelyistä esittävät esimerkiksi Elmasri ja Navathe [Elmasri ja Navathe 2004].

Rakenteistamisen lopputuotteena on XML-merkattu muoto aineistosta, jossa kirjaukset ja lauseet erotellaan koneluettavalla tavalla toisistaan. Kirjauselementteihin lisätään myös mahdollisuuksien mukaan kirjauksen aikaleimaa ja kirjaajaa kuvaavat attribuutit.

3.3.1 Kirjausten erottaminen

Aineiston siistimisen ja anonymisoinnin jälkeen aineisto jäsennetään kirjauksiksi. Kirjauksille pyritään samalla myös tunnistamaan kirjaaja sekä päivämäärä. Kirjaajien ja kirjausrajojen tunnistamisessa hyödynnetään anonymisointivaiheessa tuotettua tietoa kunkin tapauksen nimisanoista. Alla on listattu muutamia esimerkkejä eri tavoista aloittaa kirjaus:

- `151204/Seppo Nenonen`
- `20.1.05/Kaisa Käki/Lastensuojelupäivystäjä`
- `28.1.05`
- `17.8.09 Tavattu Justiina tstolla. Käydään läpi kesän kuulumiset. Justiinan tavannut`
- `27.9.06 Avotyön kotikäynti/ Kaisa Mäkinen/ Hilkka Summa. Paikalla Niko-Petteri ja`
- `020606/Paussi`
- `28.04.2009 /Jari Tölkkimäki`

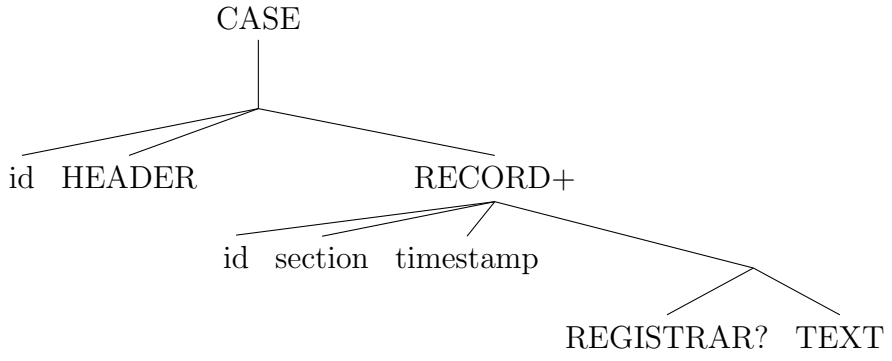
Tyypillisesti kirjaus alkaa joko pitkällä tai lyhyellä aikaleimalla, jota seuraa kirjaajan nimi tai kirjauksen tehnyt taho. Useimmiten nämä tiedot ovat myös omalla rivillään, mutta kuten edellä on todettu, tämä ei päde kaikkien kirjaajien kohdalla. Varsin säännönmukaisesti myös tapauksen lopussa on kirjattuna kirjaajan nimi omalla rivillään.

Kirjausten rajojen päättelyssä käytetään apuna edellä kerrottuja havaintoja. Lisäksi nimisanatietoa käytetään kirjaajan nimen tunnistamisessa. Vaihtoehtoisten aloitustapojen monimuotoisuudesta huolimatta pystytään kirjausten rajat tunnistamaan melko hyvin. Lisäksi rajoja voidaan merkitä myös manuaalisesti. Manuaalisesti kirjausrajoja on lisätty tapauksiin yhteensä 167 kappaletta, kirjausten kokonaismäärän ollessa 2220 kirjausta.

3.3.2 Kirjausten muuntaminen karkeaan XML-muotoon

Kun kirjausten rajat on merkitty tekstin lomaan, muunnetaan tapaukset XML-muotoon. Tätä ensimmäistä XML-muotoa kutsutaan karkeaksi XML-muodoksi, koska siinä pelkät kirjaukset erotetaan toisistaan ja niiden sisältöä käsitellään yhtenä yksikkönä. Kuvassa 3.4 on esitetty karkean XML-muodon puurakenne. Elementit on esitetty suuraakkosin ja attribuutit pienaakkosin. Elementin nimen lopussa oleva merkintä '+' tarkoittaa vähintään yhtä elementin esiintymää ja merkintä '?' tarkoittaa elementin yksittäistä valinnaista esiintymää.

Kuva 3.4 Esimerkki karkean XML-tallennusmuodon hierarkkisesta rakenteesta.



Kuvassa 3.5 on DTD tätä karkeaa XML-tallennusmuotoa varten. Aineiston karkeassa XML-muotoisessa esityksessä juurielementtinä on *case*-elementti, rivi 3. Juuren alla on *header*-elementti, joka sisältää ennen varsinaisia kirjauksia olevan tekstin, rivi 4. Tyypillisesti *header*-elementti sisältää pääasiassa henkilö- ja yhteystietoja sekä mahdollisia sosiaaliseen verkostoon kuuluvia henkilöitä. Lisäksi juurielementin lapsena on joukko *record*-elementtejä, jotka sisältävät varsinaiset kirjaukset. Kukin kirjaus on siis oman *record*-elementtinsä sisällä, rivi 5. Jokaisella kirjauksella on attribuuttina tunniste, aikaleima, sekä tieto kirjauksen osiosta, rivit 10-12. Mikäli aikaleimaa ei ole tunnistettu, on *timestamp*-attribuutin arvona 0.

Kuvassa 3.6 on suppea esimerkki karkeaan XML-muotoon tallennetusta aineistosta. Esimerkki noudattaa kuvassa 3.5 kuvattu DTD-määrittelyä. Alussa olevaan header-elementtiin on koottu ennen ensimmäistä kirjausta oleva teksti, joka tyypillisesti sisältää tapauksen esimerkissäkin kuvatut otsikkotiedot. Osassa tapauksista header-elementti sisältää myös sosiaalityöntekijöiden kirjaamia yhteystietoja tapaukseen liittyen.

Kuva 3.5 Kirjausten karkean tallennusmuodon DTD.

```
<!DOCTYPE CASE [

<!ELEMENT CASE (HEADER, RECORD+)>
<!ELEMENT HEADER (#PCDATA)>
<!ELEMENT RECORD (REGISTRAR?, TEXT)>
<!ELEMENT REGISTRAR (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>

<!ATTLIST CASE ID CDATA #REQUIRED>
<!ATTLIST RECORD ID CDATA #REQUIRED>
<!ATTLIST RECORD SECTION CDATA #REQUIRED>
<!ATTLIST RECORD TIMESTAMP CDATA #REQUIRED>
]>
```

Kuva 3.6 Esimerkki karkeasta XML-tallennusmuodosta.

```
<?xml version='1.0' encoding='UTF-8'?>
<case id="1">
  <header>
    ASIAKASKERTOMUS
    Lastensuojelu
    11.8.2011

    Feynman, Pilvi Hilma
    Kuusitie 6

  </header>
  <record id="5"
          section="AVOHUOLLON SOSIAALITYÖN KERTOMUSOSIO"
          timestamp="2005-05-17">
    <registrar>Kantola Eerika</registrar>
    <text>Tehty päätös avotyöstä 10.5.2005 alkaen.
    Kantola Eerika
    </text>
  </record>
</case>
```

3.3.3 Kirjausten pilkkominen lauseiksi

Esikäsittelyn lopuksi kirjaukset pilkotaan lauseiksi. Tämä voitaisiin suorittaa myös vasta jäsennysvaiheessa, mutta tässä työssä se tehdään tietosuojasyistä paikallisesti osana esikäsittelyä. Tietosuojaan liittyviä seikkoja on käsitelty osiossa 2.1.

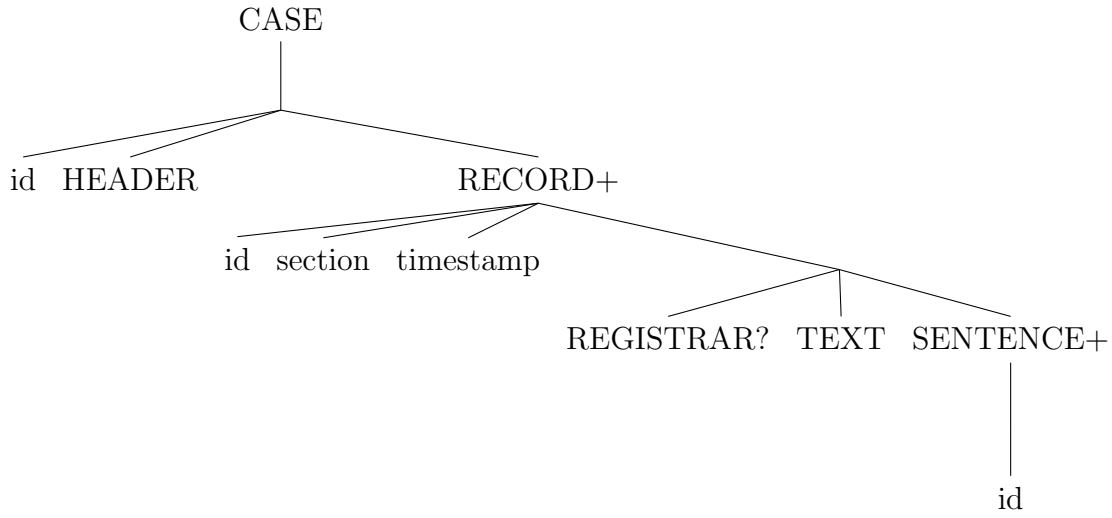
Kirjausten pilkkomiseksi lauseisiin käytetään Natural Language Toolkit -ohjelmistopaketin [Bird, Loper ja Klein 2009] Punkt-lauseidenerottajaa (sentence tokenizer). Käytetyn lauseidenerottajan toiminta pohjautuu ohjaamattomaan oppimiseen [Kiss ja Strunk 2006]. Näin ollen se pyrkii itse oppimaan esimerkiksi pisteisiin päättyvät lyhenteet, jotka eivät kuitenkaan ole lauserajoja [Kiss ja Strunk 2006]. Erottaja tukee useita eri kieliä, sillä sen käyttämää mallia voidaan vaihtaa [Nothman et al. 2012]. Tässä työssä käytetään erottelijan Jan Strunkin suomenkielisellä aineistolla opettamaa mallia [2009]. Malli on opetettu *Suomen kielen tekstikokoelman* avulla, joka koostuu lehtiartikkeleista, [FTC 2000]. Asiakaskertomusaineiston ja opetusaineistona käytetyn tavanomaisen lehtiartikkelitekstin välillä on eroavaisuuksia. Lauseiden erottamisen kannalta olennaisia eroja ovat muun muassa erilaiset lyhenteet sekä lastensuojeluaineistolle ominaisempi ajoittainen luettelonomaisuus liittyen esimerkiksi päivämääriin. Lisäksi erona on aineiston ajoittainen kielellinen huolimattomuus, joka oletettavasti johtuu työntekijöiden kiireestä, sekä lehtiartikkeleihin verrattuna erilaisesta tekstintuottamisen ja tarkastamisen prosessista. Sataa satunnaista eroteltua lausetta tarkastelemalla havaittiin, että väärin eroteltuja lauseita oli viisi kappaletta. Väärin erotellut lauseet on listattu alla:

- `Haastattelusta kävi ilmi, että Henna on ollut sijoitettuna n.`
- `24.10.`
- `Taysin lasten psyk.`
- `(Tukiryhmän työntekijät kävivät aamupäivällä kertomassa uudesta perhehoidon toimintatavasta.`
- `Aktissa.`

Suppean tarkastelun perusteella lauseiden erottelu onnistui varsin hyvin. Parempiin tuloksiin voitaisiin mahdollisesti päästä käyttämällä Punkt-lauseidenerottajan opetusaineistona asiakaskertomuksia, mutta tätä ei tässä työssä ole tehty. Nykyisen asiakaskertomusaineiston käyttämiseen liittyy myös kysymys siitä, onko aineisto tarpeeksi laajaa ohjaamatonta oppimista varten.

Erotetut lauseet lisätään uusina elementteinä kohdassa 3.3.2 kuvattuun XML-rakenteeseen. Lausetasolle laajennetussa määrittelyssä *record*-elementin lapsiksi lisätään joukko *sentence*-elementtejä, jotka sisältävät kirjauksen tekstin lause kerrallaan. Kirjauksen alkuperäinen teksti säilytetään *text*-elementissä jatkokäyttöä varten. Päivitetyn, lausetasolle rakenteistetun, XML-muodon puurakenne on kuvassa 3.7. Puurakenteeseen liittyvä DTD-määrittely on esitetty kuvassa 3.8.

Kuva 3.7 Esimerkki lausetason XML-tallennusmuodon hierarkkisesta rakenteesta.



Kuva 3.8 Lausetason XML-tallennusmuodon DTD-määrittely.

```
1  <!DOCTYPE CASE [
2
3  <!ELEMENT CASE (HEADER, RECORD+)>
4  <!ELEMENT HEADER (#PCDATA)>
5  <!ELEMENT RECORD (REGISTRAR?, TEXT, SENTENCE+)>
6  <!ELEMENT REGISTRAR (#PCDATA)>
7  <!ELEMENT TEXT (#PCDATA)>
8  <!ELEMENT SENTENCE (#PCDATA)>
9
10 <!ATTLIST CASE ID CDATA #REQUIRED>
11 <!ATTLIST RECORD ID CDATA #REQUIRED>
12 <!ATTLIST RECORD SECTION CDATA #REQUIRED>
13 <!ATTLIST RECORD TIMESTAMP CDATA #REQUIRED>
14 <!ATTLIST SENTENCE ID CDATA #REQUIRED>
15 ]>
```

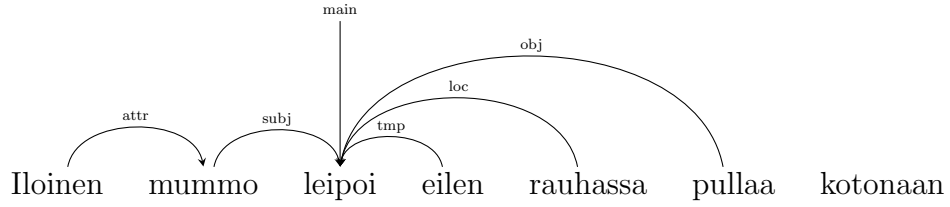
3.4 Jäsentäminen

Kirjauksiin ja lauseisiin jaettu aineisto jäsennetään dependenssijäsentimellä. Dependenssijäsennin noudattaa dependenssikielioppia, jota kutsutaan myös kemiaan viittaavalla nimellä valenssiteoria [Tarviainen 1977]. Sanan valenssilla tarkoitetaan sen pakollisten määreiden lukumäärää. Valenssi koskee erityisesti verbejä, mutta myös esimerkiksi substantiiveja. [Karlsson 2008] Tässä yhteydessä valenssi on verbien kohdalla "kyky vaatia tiettyä määrää tietyntyyppisiä määritteitä, jotta syntyisi kieliopillisesti täydellinen tai moitteeton lause" [Tarviainen 1977]. Esimerkiksi lauseessa "Hän lukee kirjaa", pääverbi *lukee* vaatii sopivalla tavalla taivutetun, oikeaa sanaluokkaa edustavan sanan subjektiksi ja objektiksi, ollakseen kieliopillisesti oikein. Suomen kielessä verbit vaativat useimmiten nollasta kolmeen määritettä [Karlsson 2008]. Säähän liittyvät verbit *sataa* ja *tuulee* ovat esimerkkejä nollapaikkaisista verbeistä. Aiemman esimerkin *lukee* verbi taas vaatii kaksi määritettä.

Dependenssijäsennyksessä lause jäsennetään pääverbistä käsin. Pääverbille tunnistetaan sen valenssin vaatimat määritteet, kuten subjekti, objekti ja erilaiset toiminnan tapaa tai aikaa kuvaavat määritteet. Myös substantiiveille tunnistetaan niitä määrittävät sanat. Lopputuloksena lauseelle saadaan jäsennyspuu, jossa sanojen väliset riippuvuudet ja niiden tyypit on esitetty. Kuvassa 3.9 on kuvattu dependenssijäsentimen jäsentämän lauseen dependenssirakenne. Sanaluokkia ja muita jäsentimen tuottamia tietoja ei ole kuvattu. Esimerkkilauseena on: "Iloinen mummo leipoi eilen rauhassa pullaa kotonaan". Lauseen pääsana on sen pääverbi *leipoa*. Subjektina ja objektina taas ovat sanat *mummo* ja *pulla*. Adjektiivi *iloinen* määrittää sanaa *mummo*. Esimerkissä sana *rauhassa* on tunnistettu virheellisesti paikan ilmaukseksi sanan *kotonaan* sijaan. Oikeassa jäsennyksessä sana *rauhassa* olisi tunnistettu tavan määreeksi ja *kotonaan* paikan määreeksi. Virheellisestä jäsennyksestä johtuen *kotonaan* sanalle ei ole päätelty roolia suhteessa muihin sanoihin. Esimerkki kuvaa varsin hyvin käytetyn dependenssijäsentimen kykyä jäsentää lauseita. Pääosin jäsennys onnistuu, mutta myös virheitä ja puutteita ilmenee. Käytetyn aineiston ongelmana jäsennyksessä on paikoitellen haastava ja vajavainen lauserakenne, ajoittainen puhekielisyys sekä sanasto, jota jäsennin ei tunne, esimerkiksi lyhenteet ja ammattitermit.

Jäsentämiseen käytetään Connexor Oy:n fi-fdg -dependenssijäsennintä [Tapanainen ja Järvinen 1997]. Se on morfologinen ja syntaktinen jäsennysohjelma suomenkieliselle tekstille. Kuvassa 3.10 on jäsentimen tuloste sen jäsennettyä lauseen: "Tarkistetaan jatko". Jäsennin palauttaa jäsennetyt lauseet XML-muodossa. Tulostetta esitellään tarkemmin osiossa 3.4.1. Jäsentämisen jälkeen XML-muotoinen data muunnetaan RDF-muotoon, jotta sitä olisi helpompi käsitellä. RDF-muodon tarkempi kuvaus on osiossa 4.2.

Kuva 3.9 Esimerkki dependenssijäsennyksestä



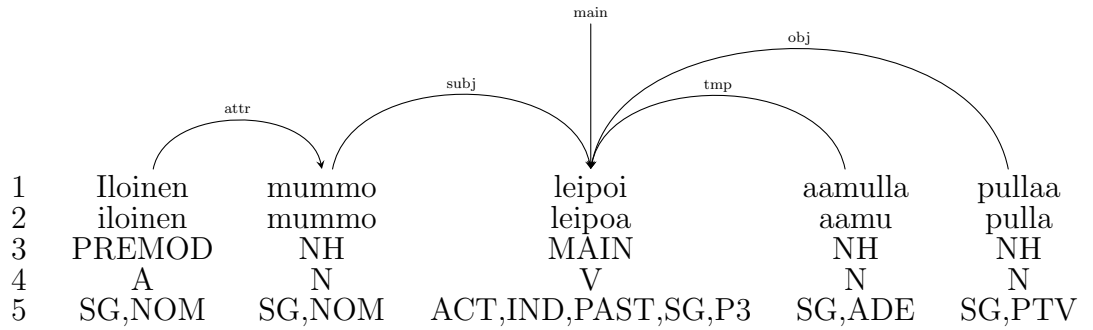
Kuva 3.10 Esimerkki fi-fdg -dependenssijäsentimen tulosteesta.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE analysis SYSTEM "http://www.conexor.fi/dtds/3.7/fdg3.dtd">
<analysis>
  <sentence id="w1">
    <token id="w2">
      <text>Tarkistetaan</text>
      <lemma>tarkistaa</lemma>
      <depend head="w1">main</depend>
      <tags>
        <syntax>@MAIN</syntax>
        <morpho>V PASS IND PRES</morpho>
      </tags>
    </token>
    <token id="w3">
      <text>jatko</text>
      <lemma>jatko</lemma>
      <depend head="w2">obj</depend>
      <tags><syntax>@NH</syntax>
      <morpho>N SG NOM</morpho>
      </tags>
    </token>
  </sentence>
</analysis>
```

3.4.1 Jäsentimen tuloste

Tässä osiossa kuvataan dependenssijäsentimen tulosteen sisältöä. Kuvassa 3.10 on esimerkki fi-fdg:n XML-muotoisesta tulosteesta ja kuvassa 3.11 on kuvattu jäsentimen tulostetta jäsennyspuumuodossa. Nimetyt kaaret esittävät sanojen välisiä riippuvuuksia. Ensimmäisellä rivillä on esimerkkilause: “Iloinen mummo leipoi aamulla pullaa”. Sanojen lemmat ovat toisella rivillä. Kolmannella rivillä on kuvattu sanojen syntaktiset funktiot. Sanojen sanaluokat on esitetty neljännellä rivillä. Viimeisellä rivillä on listattu sanan taivutukseen liittyvää tietoa.

Kuva 3.11 Laajennettu esimerkki dependenssijäsennyksestä.



Jäsennin palauttaa tulosteenaan jäsentämänsä lauseen sekä jokaiselle jäsennetylle sanalle analyysin. Sana ja sen lemma esitetään käyttäen *text*- ja *lemma*-elementtejä. *Depend*-elementissä kuvataan sanan mahdollisen dependenssin tyyppi ja kohde. Riippuvuuskohde on aina toinen sana, paitsi lauseen pääverbillä, joka riippuu lause-elementistä ja jonka riippuvuus on tyyppiä *main*. Muita riippuvuustyyppejä on listattu taulukossa 3.3. Riippuvuustyyppejä on yhteensä 31 kappaletta. Täydellinen lista riippuvuuksista löytyy jäsentimen dokumentaatiosta [Connexor 2006]. Taulukossa on listattu tässä työssä pääasiallisesti hyödynnettyjä riippuvuuksia.

Taulukko 3.3: Sanojen välisten riippuvuuksien tyyppejä

Riippuvuuden nimi	Riippuvuuden kuvaus
main (main verb)	lauseen pääverbi
subj (subject)	verbin subjekti
obj (object)	verbin objekti
attr (attribute)	sanan attribuutti
sou (source)	kuvaa toiminnan lähdettä tai aihetta
tmp (temporal)	kuvaa toiminnan ajallista määrettä

Sanan syntaktinen funktio kuvaa sen tehtävän lausekkeessa. Sanan syntaktisesta funktiosta käytetään myös nimitystä lauseenjäsen. [Karlsson 2008] Jäsennin esittää sanan päätellyn syntaktisen funktion lausekkeessa *syntax*-elementissä. Taulukossa 3.4 on esitetty työn kannalta tärkeimpien syntaktisia funktiota. Esimerkkejä näistä funktioista ovat *substantiivilausekkeen pääsana* tai *substantiivilausekkeen etumäärite*. Pääsana on usein substantiivi ja etumäärite sitä kuvaava sana, esimerkiksi adjektiivi. Kuvassa 3.11 rivillä kolme on esitetty sanojen syntaktiset funktiot. Kuten kuvasta taulukon 3.4 avulla havaitaan on substantiivilausekkeen "iloinen mummo" pääsanana *mummo* ja sen etumäärite on sana *iloinen*. Koko substantiivilauseke taas toimii sanojen riippuvuuksia kuvaavien kaarien mukaisesti verbin *leipoa* subjektina.

Taulukko 3.4: Sanojen syntaktisia funktioita

Syntaktisen funktion nimi	Syntaktisen funktion kuvaus
NH (noun head)	substantiivilausekkeen pääsana
PREMOD (premodifier)	substantiivilausekkeen etumäärite
POSTMOD (postmodifier)	substantiivilausekkeen jälkimäärite

Sanojen sanaluokkia jäsennin merkitsee taulukon 3.5 mukaisesti. Taulukossa on vain osa jäsentimen sanaluokkatunnuksista. Jäsentimen tulosteessa sanaluokka on tallennettu *morpho*-elementissä. Sanaluokan lisäksi jäsennin esittää tietoa sanan taivutuksesta *morpho*-elementissä. Taulukossa 3.6 on listattu muutamia jäsentimen käyttämiä morfologisia merkintöjä.

Taulukko 3.5: Sanaluokkien lyhenteet

Sanaluokan lyhenne	Sanaluokan nimi
V (verb)	verbi
N (noun)	substantiivi
A (adjective)	adjektiivi

Taulukko 3.6: Sanaluokkien lyhenteet

Morfologinen lyhenne	
SG (singular)	yksikkömuoto
PL (plural)	monikkomuoto
GEN (genitive)	genetiivi
NOM (nominative)	nominatiivi
PTV (partitive)	partitiivi
ADE (adessive)	adessiivi

3.5 Annotointi

Jäsennettyä ja RDF-muotoon siirrettyä aineistoa rikastetaan annotoinneilla. Osaan aineiston sanoista lisätään sopiva annotaatio, jota hyödynnetään myöhemmissä työvaiheissa. Annotointi tapahtuu kahdessa osassa. Ensimmäisessä vaiheessa aineisto käydään kertaalleen läpi ja siihen lisätään sana-annotaatiot. Toisessa vaiheessa taas käytetään hyväksi ensimmäisen vaiheen annotaatioita uusien annotaatioiden luomiseen.

3.5.1 Ensimmäinen annotaatiokierros

Tunnistetut päivämäärät, luvut, kellonajat ja nimisanat annotoidaan. Luvut ja ajanilmaukset annotoidaan, jotta ne olisi helppo sulkea myöhemmissä analyyseissä käsittelyn ulkopuolelle. Aineistosta löydetyt nimet annotoidaan joko etu- tai sukunimiksi. Anonymisointivaiheessa tuotettuja korvaussääntöjä käytetään tässä kohtaa nimisanojen tunnistamiseen.

Teknisesti annotointi tapahtuu vertaamalla sanoja tunnistettujen nimisanojen listaan sekä Prologilla toteutettuihin DCG-sääntöihin [Finin ja Palmer 1983]. Prologilla toteutettuja *definite clause grammar* (DCG) -sääntöjä käytetään tässä työssä jäsentämään päivämääriä ja ajanilmauksia. Koska osana sääntöjen soveltamista voidaan vapaasti suorittaa laskentaa, on tätä hyödynnetty toteutettaessa ajanilmauksia tunnistavia jäsennyssääntöjä, jotka ilmauksen muodon lisäksi tarkistavat ajanilmauksen oikeellisuuden. Esimerkkinä tästä voidaan käyttää ajanilmauksia ‘12.13’, ‘12.10’ ja ‘32.10’. Näistä ensimmäinen on kellonaika, toinen on kelloaika tai päivämäärä ja kolmas ei ole mikään edellisistä. Päivämäärät ja kellonajat annotoidaan eri tavoin ja epäselvät ilmaukset, kuten ‘12.10’, annotoidaan tarkemmin määrittelemättömiksi ajanilmauksiksi.

3.5.2 Toinen annotaatiokierros

Jälkimmäisellä annotointikierroksella jatkokäsitellään edellisellä kierroksella tehtyjä nimiannotaatioita. Aineistoon tehdyt nimiannotaatiot käydään läpi ja yksittäiset nimisa-

naannotaatiot pyritään yhdistämään kokonaisiksi nimiksi. Kokonaisten nimien päättelyssä oletetaan peräkkäisen etu- ja sukunimiparin liittyvän yhteen ja samaan henkilöön. Toisin sanoen peräkkäinen etu- ja sukunimi liitetään uudella annotaatiolla yhdeksi kokonaisuudeksi.

Annotointia jatketaan tulkitsemalla tähän mennessä tunnistetut nimisanakokonaisuudet ja jäljellä olevat yksittäiset nimisanat jonkin nimen ilmentymiksi. Nämä ilmentymät pyritään liittämään niiden viittauskohteeseen eli tapauksen henkilön kokonaiseen nimeen. Taidemaalarien elämäkertoja WWW-sivuilta koostava ArtEquAKT-järjestelmä [Weal et al. 2007] on onnistuneesti käyttänyt nimien samankaltaisuutta heuristiikkana yhdistellessään eri lähteistä kerättyä tietoa. Myös tässä työssä hyödynnetään samaa heuristiikkaa. Toisen annotaatiokierroksen lopputuloksena on siis joukko henkilöiden kokonaisia nimiä, joihin on liitetty kaikki niiden tunnistetut ilmentymät aineistossa. Yksittäinen ilmentymä voi koostua yhdestä tai kahdesta nimisanasta. Mikäli ilmentymään kuuluu pelkkä etu- tai sukunimi, pyritään sille löytämään sitä vastaava kokonimi. Tieto ilmentymistä ja ilmentymien viittauskohteista tallennetaan edelleen annotaatioina aineistoon.

4 Jäsennetyn ja annotoidun aineiston tallennusmuoto

Jäsentämisen jälkeen aineisto muunnetaan RDF-muotoon. Myös tehdyt annotaatiot liitetään osaksi aineistoa RDF-muodossa. Tässä osiossa kuvaillaan RDF-kieltä yleisesti sekä käydään läpi sen käyttöä jäsennetyn tekstin tallennusmuotona.

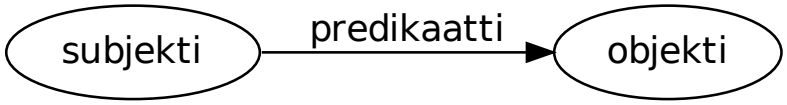
RDF valikoitui tallennusmuodoksi graafitietomallinsa perusteella, koska jäsennettyjen lauseiden esittäminen ja käsittely on sen avulla varsin luontevaa. Aineiston tallentaminen sanagraafina helpottaa useita työvaiheita. Esimerkiksi henkilöiden roolien päättely voidaan toteuttaa graafin täsmäyksenä (graph matching). Roolien täsmäyksen toteutusta on kuvattu tarkemmin osiossa 5.2.1. Graafimallin avulla tekstin sisältöä ja sanojen keskinäisiä suhteita on luontevampaa lähestyä verrattuna lineaariseen sanajonoon. Dependenssijäsennetyt, toisiinsa viittaavista sanoista koostuvat lauseet on luontevaa esittää verkostona ilman hierarkiaa.

RDF eli Resource Description Framework on alunperin kieli WWW-resursseihin liittyvän tiedon esittämistä varten. Se on erityisesti tarkoitettu WWW-resurssien, kuten WWW-sivujen, metadatan esittämiseen. Sillä voidaan siis ilmaista esimerkiksi WWW-sivun otsikko, tekijä tai luontiaika. [E. Miller ja Manola 2004]. RDF on suunniteltu esittämään informaatiota joustavalla ja mahdollisimman vähän rajoitteita sisältävällä tavalla [Carroll ja Klyne 2004]. Esitystapa on kuitenkin sellainen, että sitä on mahdollista käsitellä koneellisesti [E. Miller ja Manola 2004].

4.1 RDF-tietomalli

RDF-tietomallissa tieto esitetään subjekti-predikaatti-objekti -kolmikoina (triple). Kolmikoita voidaan kutsua myös faktoiksi, sillä jokainen kolmikko kuvaa subjektin olevan predikaatin määrittämässä suhteessa objektiin. Kuvassa 4.1 on esimerkki RDF-graafista, jossa on yksi kolmikko. Graafin solmuja ovat subjektit ja objektit, kun predikaatit taas kuvataan kaarina. Kaaret ovat aina suunnattuja ja osoittavat kohti objektia. [Carroll ja Klyne 2004]

Kuva 4.1 Esimerkki RDF-kolmikosta.



Seuraavaksi määritellään tietomalliin liittyvät käsitteet tarkemmin. Määrittelyjen lähteinä on käytetty W3C:n suosituksia [E. Miller ja Manola 2004] [Carroll ja Klyne 2004].

RDF URI on merkkijono, joka identifioi fyysisen tai abstraktin resurssin. Resurssi voi olla mikä tahansa olio, jolla on identiteetti, kuten elektroninen dokumentti, kuva tai kokoelma resursseja. Myös fyysiset tai abstraktit objektit kuten kirjat, yritykset tai ihmiset voivat olla resursseja, vaikka ne eivät olisikaan saatavissa Internetin välityksellä. [Berners-Lee, Fielding ja Masinter 2005]

RDF-literaali, lyhemmin pelkkä literaali, voi olla tyypitetty tai tavallinen (plain). Lyhyesti voidaan todeta, että RDF-literaali on joko merkkijono tai jonkin tietotyypin mukainen arvo, kuten kokonaisluku. Literaali koostuu yhdestä tai kahdesta nimetystä osasta ja se voi esiintyä vain objektina. Molempien literaalityyppien ensimmäinen ja pakollinen osa on leksikaalinen muoto (lexical form), joka on Unicode-merkkijono. Tavallisilla literaaleilla on tämän lisäksi valinnainen kielitunniste (language tag). Tyypitetyillä literaaleilla taas on pakollinen tietotyypin URI, joka määritellään kuten RDF URI.

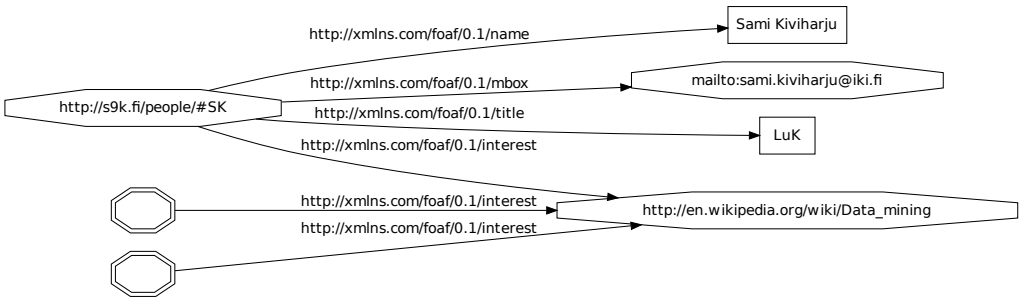
Anononyymi solmu käyttäytyy muuten samalla tavalla kuin RDF URI, mutta se ei identifioi mitään resurssia. Anonyymeillä solmuilla on kuitenkin identiteetti, joten ne voidaan erottaa toisistaan. Mikäli RDF URI:n ajatellaan viittaavan johonkin tiettyyn resurssiin, voidaan anonyymin solmun ajatella viittaavan tarkemmin määrittelemättömään resurssiin.

RDF-kolmikko (RDF triple) koostuu kolmesta osasta:

- subjektista, joka on *RDF URI* tai *anonyymi solmu*,
- predikaatista, joka on *RDF URI* ja
- objektista, joka on on *RDF URI*, *literaali* tai *anonyymi solmu*.

RDF-graafi muodostuu joukosta *RDF-kolmikoita*.

Kuva 4.2 Esimerkki RDF-graafista



Kuvassa 4.2 on kuvattu esimerkki RDF-graafista, joka noudattaa The Friend of a Friend (FOAF) -ontologiaa. FOAF-ontologia on tarkoitettu etenkin henkilöihin liittyvän metadatan kuvaamiseen, jota tarkoitusta varten se tarjoa valmiin sanaston RDF-kieltä varten. [Brickley ja L. Miller 2010]

Kuvan 4.2 graafi kuvaa RDF URI -resurssiin **http://s9k.fi/people/#SK** liittyvän seuraavanlaisia FOAF-ontologian mukaisia predikaatteja:

- *name*, objektinaan RDF-literaali,
- *mbox*, objektinaan sähköpostiosoitetta kuvaava RDF URI,
- *title*, objektinaan RDF-literaali,
- *interest*, objektinaan RDF URI.

Graafin kahdeksankulmiot ovat siis RDF URI -tyyppisiä ja RDF-literaalit on esitetty suorakulmioina. Edellisten lisäksi graafissa on myös kaksi anonyymiä solmua, jotka on esitetty kaksoiskahdeksankulmioina. Niillä ei ole viittauksen kohteena olevaa resurssia, mutta ne voidaan silti erottaa toisistaan ja niille voidaan myös RDF URI:en tapaan antaa ominaisuuksia predikaattien avulla, kuten graafissakin on tehty.

4.2 Aineisto RDF-muodossa

Kukin aineiston sana tallennetaan omana anonyymina solmunaan. Samoin tehdään lauseille, kirjauksille ja kaikille muille sovellusalueen olioille. Näin ollen sovellusalueen olioita ovat:

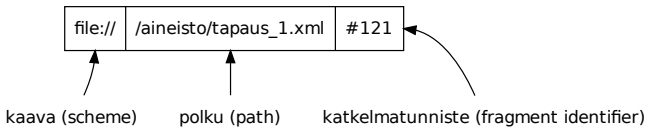
- sanat,
- lauseet,
- kirjaukset,
- nimien instanssit ja
- nimet.

Nämä listatut sovellusalueen oliot esiintyvät pääasiassa RDF-kolmikkojen subjekteina. Poikkeuksen tähän tekevät predikaatit, joissa määritellään olioiden keskinäisiä suhteita. Käytännössä anonyymejä solmuja käytetään tässä työssä abstrakteina olioina, joille lisätään ominaisuuksia.

Anonyymit solmut valikoituivat esittämään sovellusalueen olioita niiden helppouden vuoksi. Niiden tilalla olisi voitu käyttää myös RDF URL:ja, jolloin ne olisivat viitanneet edustamaansa olioon sopivalla tavalla XML-merkatussa aineistossa. Esimerkiksi kirjaukset olisi voitu esittää seuraavanlaisina URL-osoitteina: file:///aineisto/tapaus_-1.xml#121. Osoitteen polkuosa osoittaisi resurssin, jossa kirjaus on tallennettuna. Katkelmatunniste (fragment identifier) taas osoittaisi tarkemman kohdan XML-merkatussa tiedostossa ja vastaisi käytännössä jonkin id-attribuutin arvoa. Kuvassa 4.3 on havainnollistettu URL-osoitteen osia [Berners-Lee, Fielding ja Masinter 2005]. Samaa linkitystapaa voitaisiin soveltaa myös sanoihin, jos ne merkattaisiin kukin oman element-

tinsä sisään. Tämän hetkisessä XML-tallennusmuodossa voidaan suoraan viitata vain kirjauksiin ja lauseisiin, koska vain niillä on omat tunnisteet.

Kuva 4.3 URL-osoitteen osia



4.2.1 Sanoihin liittyvät predikaatit

Kaikilla anonyymeillä solmuilla on *isa*-predikaatti, joka määrittää solmun tyypin. Anonyymeja solmuja, joilla *isa*-predikaatin objektina on literaali '#word', kutsutaan sanasolmuiksi. Sanasolmuihin liittyvät predikaatit on esitetty taulukossa 4.1. Predikaatit 2-7 tulevat suoraan dependenssijäsentimen tulosteesta, jota on esitelty kohdassa 3.4.1. *Text*-predikaatin avulla on esitetty sana siinä muodossa, jossa se tekstissä esiintyy. *Lemma* taas kuvaa sanan perusmuodon jäsentimen tuottamassa muodossa. Mikäli sana on yhdyssana, on sen perusmuoto ositetussa muodossa, esimerkkinä *levykauppaauto* sana esitetään muodossa *levy#kauppa#auto*. *Morpho*-predikaatin avulla esitetään sanan päätelty morfologia. *Syntax*-predikaatti kuvaa sanan syntaktisen funktion. *Depend*-predikaatin objektina on toinen sanasolmu. Sana siis viittaa siihen sanaan, josta se riippuu. Predikaatti *dep_type* kuvaa edellä mainittua sanan riippuvuuden tyyppiä. Sana liitetään lauseeseensa *part_of_sentence*-predikaatilla, jonka arvona on lausesolmu. Tämä predikaatti ei kuitenkaan määritä sanan sijaintia lauseessa vaan tähän käytetään *word_pos_in_sentence*-predikaattia, jonka arvona on kokonaisluku.

Taulukko 4.1: Sanoihin liittyvät predikaatit

#	Predikaatin nimi	Kuvaus	Predikaatin objekti
1	isa	Määrittää solmun tyypin	'#word'
2	id	Sanan tunnus fi-fdg:n tulosteessa	esim. 'w5'
3	text	Aineistossa esiintyvä sana	esim. 'kertoi'
3	lemma	Sanan perusmuoto	esim. 'kertoa'
4	morpho	Sanan morfologinen jäsennys	esim. 'V', 'NOM', 'ACT'
5	syntax	Sanan syntaktinen rooli	esim. 'ADVL', 'NH'
6	depend	Sana josta tämä sana riippuu	sanasolmu
7	dep_type	Sanan riippuvuuden tyyppi	esim. 'main'
8	part_of_sentence	Lause johon sana kuuluu	lausesolmu
9	word_pos_in_sentence	Sanan indeksi lauseen sisällä	kokonaisluku

4.2.2 Lauseisiin liittyvät predikaatit

Kuten sanasolmujen kohdalla, määrittää *isa*-predikaatti anonyymin solmun lausetta kuvaavaksi solmuksi. *serial_number*-predikaattilla esitetään lauseen järjestysnumero aineiston dokumentissa. Kirjauksiin lauseet linkitetään *part_of_record*-predikaatilla, jonka objektina on se kirjaussolmu, johon kyseinen lausesolmu liittyy. *sentence_pos_in_record*-predikaatti kuvaa lauseen järjestysnumeron siinä kirjauksessa, johon lause liittyy. Lauseisiin liittyvät predikaatit on esitetty taulukossa 4.2.

Taulukko 4.2: Lauseisiin liittyvät predikaatit

#	Predikaatin nimi	Kuvaus	Predikaatin objekti
1	isa	Määrittää solmun tyypin	'#sentence'
2	serial_number	Lauseen sarjanumero dokumentissa	kokonaisluku
3	part_of_record	Kirjaus, johon lause kuuluu	kirjaussolmu
4	sentence_pos_in_record	Lauseen järjestysnumero kirjauksessa	kokonaisluku

4.2.3 Kirjauksiin liittyvät predikaatit

Kirjauksia kuvaavien solmujen kohdalla *isa*-predikaattia käytetään kuten edellä. Kirjauksen tapahtuma-ajan kuvaamiseen käytetään predikaatteja *has_timestamp* ja *has_date*. Näistä ensimmäinen sisältää ajan sekunteina 1.1.1970 lähtien ja jälkimmäinen päivämäärän ISO 8601-muodossa, esimerkkinä 2012-12-09. Syy kahteen tapaan ilmaista aika johtuu käytettävissä olevista kirjastoista, jotka muuttavat aikoja sekuntimuodosta helpommin luettavaan muotoon ja takaisin. Lisäksi kirjaukset on triviaalia järjestää aikajärjestykseen sekuntiarvojen perusteella. Kirjauksen osio on kuvattu *part_of_section*-predikaatilla. Predikaatin objektina on osion nimi merkkijonona. Kirjauksiin liittyvät predikaatit on esitetty taulukossa 4.3.

Taulukko 4.3: Kirjauksiin liittyvät predikaatit

#	Predikaatin nimi	Kuvaus	Predikaatin objekti
1	isa	Määrittää solmun tyypin	'#record'
2	has_timestamp	Kirjauksen aikaleima	sekuntien lukumäärä 1.1.1970 lähtien
3	has_date	Kirjauksen päiväys	päivämäärä ISO 8601-muodossa
4	part_of_section	Osio johon kirjaus kuuluu	osion nimi

Esimerkki aineistosta RDF-graafimuodossa

Kuvassa 4.5 esitetty on kolmisanaisen lauseen tallentaminen RDF-muodossa. Esimerkissä käytetty lause on “Ainokin kertoi haaveistaan.”. Kuvan solmut edustavat kolmea eri tyyppistä sovellusalueen oliota: sanoja, lauseita ja kirjauksia. Eri solmutyyppejä on havainnollistettu kuvassa 4.4. Näiden lisäksi RDF-kolmikoiden objektit on esitetty solmuina ja predikaatit kaarina. Graafia luetaan siis samalla tavoin kuin aiemmin esitettyä esimerkkiä RDF-graafista kuvassa 4.2.

Kuvan 4.5 oikeassa laidassa on kolme sanasolmua sekä niihin liittyvä data. Koska kaikki sovellusalueen oliot ovat RDF-muodossa tallennettu anonyymeinä solmuina, kerrotaan solmun tyyppi *isa*-predikaatilla, jonka objekti sanasolmuilla on ‘#word’. Tämän lisäksi sanasolmuilla on dependenssijäsentimen tuottama tunniste sekä järjestysnumero lauseen sisällä. Nämä kuvataan *id*- ja *word_pos_in_sentence*-predikaateilla.

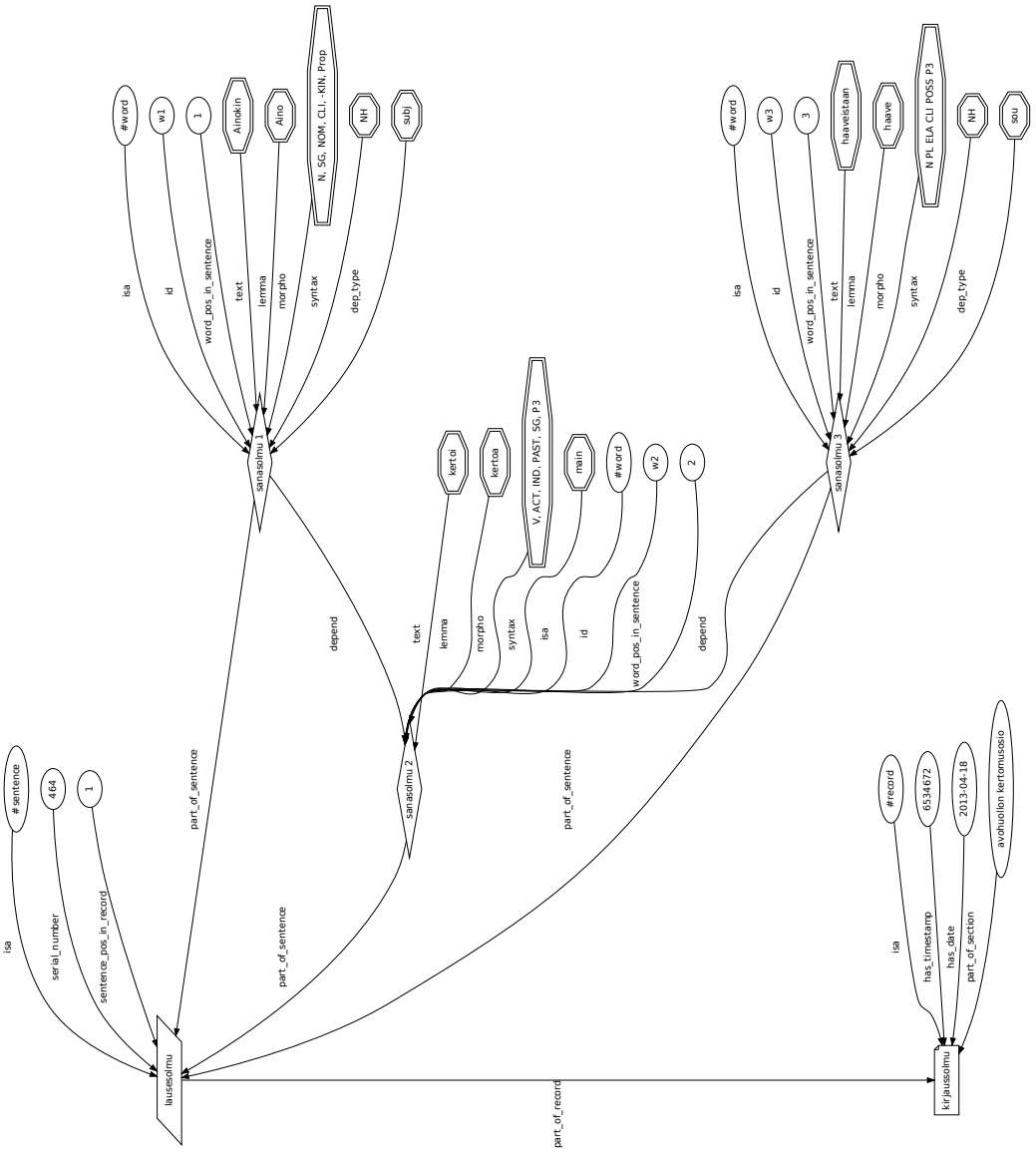
Loput sanoihin liittyvät predikaatit esittävät dependenssijäsentimen tuottamaa informaatiota. Tämä kielitieteellinen informaatio esitetään kuvassa selvyyden vuoksi solmuilla, jossa on kaksoisreunus erotuksena muista rakennetta kuvaavista objektisolmuista. Sana, siinä muodossa kuin se esiintyy aineistossa, kuvataan *text*-predikaatilla ja sanan lemma *lemma*-predikaatilla. Predikaatti *morpho* kuvaa sanan morfologiaa. Sana ‘Ainokin’ on esimerkissä jäsennetty yksikössä (sg) olevaksi nominatiivimuotoiseksi (nom) substantiiviksi (n), jolla on -kin -pääte (cli, -kin) ja jonka jäsennin päättelee erisnimeksi (Prop). Syntaktisesti ensimmäinen sana on luokiteltu substantiivilausekkeen rungoksi predikaatilla *syntax*. Sanan mahdollisen riippuvuuden tyyppi ja itse riippuvuus kuvataan *dep_type*- ja *depend*-predikaateilla. Kuvan esimerkissä ensimmäinen ja kolmas sana riippuvat molemmat lauseen toisena sanana olevasta pääverbistä ‘kertoa’. Riippuvuus esitetään kahden sanasolmun välisenä suhteena. Esimerkkilauseessa ensimmäisen sanan riippuvuuden tyypiksi on tunnistettu subjekti (subj) ja kolmannen sanan on päätelty kuvaavan aihetta (sou).

Kuvan lausesolmu on esitetty suunnikkaana ja sen tyyppi määräytyy *isa*-predikaatin arvolla ‘#sentence’. Lauseisiin liittyy lisäksi *serial_number*-predikaatilla ilmaistu juokseva numerointi. Numerointi on sama kuin lopullisessa XML-muodossa, joka on kuvattu osiossa 3.3.3. Lauseen sijainti kirjauksessa esitetään *sentence_pos_in_record*-predikaatilla.

Kirjaussolmu esitetään kuvassa suorakulmiona ja sen tyyppi määräytyy *isa*-predikaatin arvolla ‘#record’. Kirjauksen osion kuvaamiseen käytetään predikaattia *part_of_section*. Aikaleima ja päivämäärä esitetään predikaateilla *has_timestamp* ja *has_date*. Lauseet liittyvät kirjaukseen predikaatilla *part_of_record*.

Kuva 4.4 Kuvissa 4.5 ja 4.6 käytettyjen solmujen tyypit.

Kuva 4.5 Esimerkki aineistosta RDF-graafimuodossa



4.2.4 Annotaatioihin liittyvät predikaatit

Kuten aiemmissa kohdissa on käynyt ilmi, määräytyy solmun tyyppi *isa*-predikaattin avulla. Solmulle voidaan kuitenkin asettaa useita tyyppejä käyttämällä *isa*-predikaattia useampaan kertaan eri objektilla. Tällä tavoin käytettyjä objekteja ovat merkkijonot '#date', '#integer' ja '#time'. Näitä käyttämällä sanasolmuun liitetään annotaatio, mikäli sanasolmu kuvaa päivämäärää, kokonaislukua tai kellonaikaa. Annotaatioihin liittyvät predikaatit on esitetty taulukossa 4.4.

Taulukko 4.4: Annotaatioihin liittyvät predikaatit

#	Predikaatin nimi	Kuvaus	Predikaatin objekti
1	isa	Määrittää solmun tyypin	'#date'
2	isa	Määrittää solmun tyypin	'#integer'
3	isa	Määrittää solmun tyypin	'#time'

4.2.5 Nimiin liittyvät predikaatit

Nimiä käsitellään RDF-muodossa kahden eri olion avulla. Käytännössä aineistossa erotellaan henkilöiden nimet sinänsä ja niiden esiintymät tekstissä. Näin ollen solmut, jotka määritellään *isa*-predikaatin objektilla '#simple_name', edustavat henkilön kokonimeä. Tällaisia solmuja kutsutaan nimisolmuiksi. Kutakin tapauksen tunnistettua henkilöä kohti on vain yksi tällainen solmu. Jokaiseen nimeen sinänsä liittyy useita nimen esiintymiä, jotka palautuvat aineiston sanoihin. Näitä nimen esiintymä -solmuja kuvataan osiossa 4.2.6.

Predikaatit *firstname* ja *lastname* esittävät nimen etu- ja sukunimen merkkijonona. Sama tieto esitetään toteutussyistä myös Prolog-terminä predikaatilla *pl_name_term*. Termit ovat muotoa `name('Hujanen',['Toini-Aada'])` ja ne kuvaavat nimisolmut Prolog-kielelle ominaisessa muodossa. Prolog-termin sisältö on johdettua informaatiota, joten se voitaisiin tuottaa ajonaikana *firstname*- ja *lastname*-predikaattien pohjalta. Tämä johdettu informaatio kuitenkin tallennetaan ohjelmiston toiminnan nopeuttamiseksi. Nimiin liittyvät predikaatit on esitetty taulukossa 4.5.

Taulukko 4.5: Nimiin liittyvät predikaatit

#	Predikaatin nimi	Kuvaus	Predikaatin objekti
1	isa	Määrittää solmun tyypin	'#simple_name'
2	firstname	Nimeen liittyvä etunimi	etunimi merkkijonona
3	lastname	Nimeen liittyvä sukunimi	sukunimi merkkijonona
4	pl_name_term	Nimi Prolog terminä	Prolog-termi

4.2.6 Nimen esiintymiin liittyvät predikaatit

Nimen esiintymiin viitattiin jo edellisessä osiossa ja ne esitetään *isa*-predikaatin objektilla '#raw_name_instance'. Esiintymä liitetään siihen lauseeseen, jossa se esiintyy *in_sentence*-predikaatilla. Predikaatti *instance_of* liittää nimen esiintymän nimeen itseensä eli käytännössä nimisolmuun. Lauseen lisäksi esiintymä liitetään predikaateilla *firstname* ja *lastname* niihin sanojen esiintymiin eli sanasolmuihin, jotka ovat nimen varsinainen esiintymä. Predikaatilla *pl_name_n_term* nimen esiintymään liitetään kuvaus sen tietosisällöstä Prolog-terminä. Prolog-termiä käytetään kuten nimiin liittyvien predikaattien kohdalla kohdassa 4.2.5. Nimen esiintymiin liittyvät predikaatit on esitetty taulukossa 4.6.

Taulukko 4.6: Nimen esiintymiin liittyvät predikaatit

#	Predikaatin nimi	Kuvaus	Predikaatin objekti
1	isa	Määrittää solmun tyypin	'#raw_name_instance'
2	in_sentence	Lause, johon esiintymä kuuluu	lausesolmu
3	instance_of	Nimi, johon esiintymä liittyy	nimisolmu
4	firstname	Sanasolmu, joka on esiintymän etunimi	sanasolmu
5	lastname	Sanasolmu, joka on esiintymän sukunimi	sanasolmu
6	pl_name_n_term	Esiintymä Prolog terminä	Prolog-termi

Esimerkki nimien esittämisestä RDF-graafimuodossa

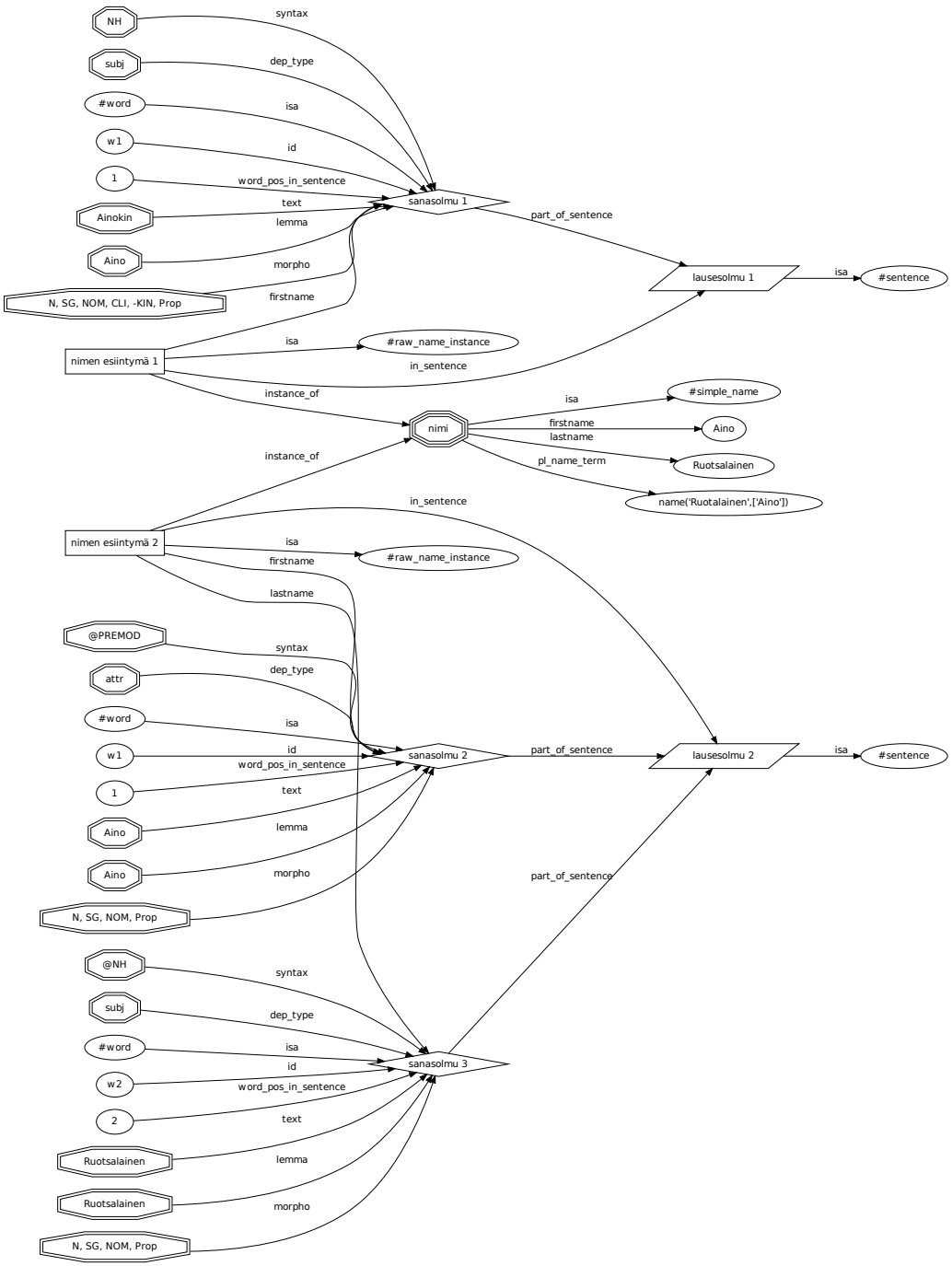
Kuvassa 4.6 on esitetty nimen *Aino Ruotsalainen* sekä sen kahden esiintymän tallentaminen RDF-muodossa. Esimerkissä nimitiedon tallentamiseen käytetään yhtä nimisolmua ja kahta nimenesiintymäsolmua. Lisäksi on kuvattu sana- ja lausesolmut, joihin edelliset viittaavat.

Henkilön nimeä kuvaavan nimisolmun tyyppi esitetään *isa*-predikaattia käyttäen arvolla '#simple_name'. Kuvassa nimisolmu on esitetty kolmoiskahdeksankulmiolla. Nimisolmuun liittyvät etu- ja sukunimi esitetään merkkijonoina käyttäen *firstname*- ja *lastname*-predikaatteja. Etu- ja sukunimitieto tallennetaan lisäksi optimointisyistä predikaatin *pl_name_term* avulla, kuten kohdassa 4.2.5 on kuvattu. Nimisolmuun itsessään ei säilötä tietoa nimen esiintymisistä aineistossa.

Tekstin tasolla nimen esiintymä on yksittäinen etu- tai sukunimi tai useampi peräkkäinen nimisana. Esiintymä voi siis koostua pelkästä etu- tai sukunimestä tai kokonimestä. Jokaiselle nimen esiintymälle luodaan oma nimenesiintymäsolmu. Tämän solmun tyyppi määritellään *isa*-predikaatin arvolla '#raw_name_instance' ja solmut on kuvattu suorakulmioina. Esiintymään liittyvä etunimi kuvataan *firstname*-predikaatilla, jonka arvona on sanasolmu. Sukunimi kuvataan samalla tavalla käyttäen predikaattia *lastname*. Esiintymäsolmuun voi liittyä useita etunimiä, mutta esiintymän olemassaololle riittää yksikin etu- tai sukunimi. Kuvassa ensimmäiseen nimen esiintymään liittyy yksi etunimi ja toiseen esiintymään liittyy sekä etu- että sukunimi.

Alkuperäisessä tekstissä ensimmäinen nimen esiintymä on sanasolmun *text*-predikaatin mukaan 'Ainokin'. Toiseen nimen esiintymään liittyvä tekstikatkelma on sanasolmujen perusteella 'Aino Ruotsalainen'. Esiintymien on kuitenkin päätelty liittyvän samaan nimisolmuun. Nimien esiintymien viitekohteiden päättely on esitelty luvussa 3.5.2.

Kuva 4.6 Esimerkki nimien ilmaisemisesta RDF-muodossa



5 Analyysi

Aineiston käsittelyn varsinaisena tarkoituksena on muodostaa alkuperäisestä aineistosta helpommin ja nopeammin ymmärrettäviä koosteita. Koosteiden avulla pyritään nostamaan esille aineiston tärkeitä kohtia sekä luomaan havainnollisia visuaalisia kuvaajia. Osa aineistosta eristetystä informaatiosta ei sellaisenaan vielä täysin palvele sosiaalityöntekijöitä, mutta ne toimivat mielenkiintoisina lähtökohtina jatkotarkastelulle. Esimerkki tällaisesta jatkokäsittelyyn soveltuvasta koosteesta on myöhemmin esiteltävä henkilöesiintymägraafi, jonka yksinkertaistettu muoto henkilöaikajana on. Näistä henkilöaikajana soveltuu kuitenkin oletettavasti paremmin sosiaalityöntekijöiden nykyisiin tiedontarpeisiin sillä se esittää henkilöesiintymägraafin tiedot karsitummassa ja jäsennetymmässä muodossa.

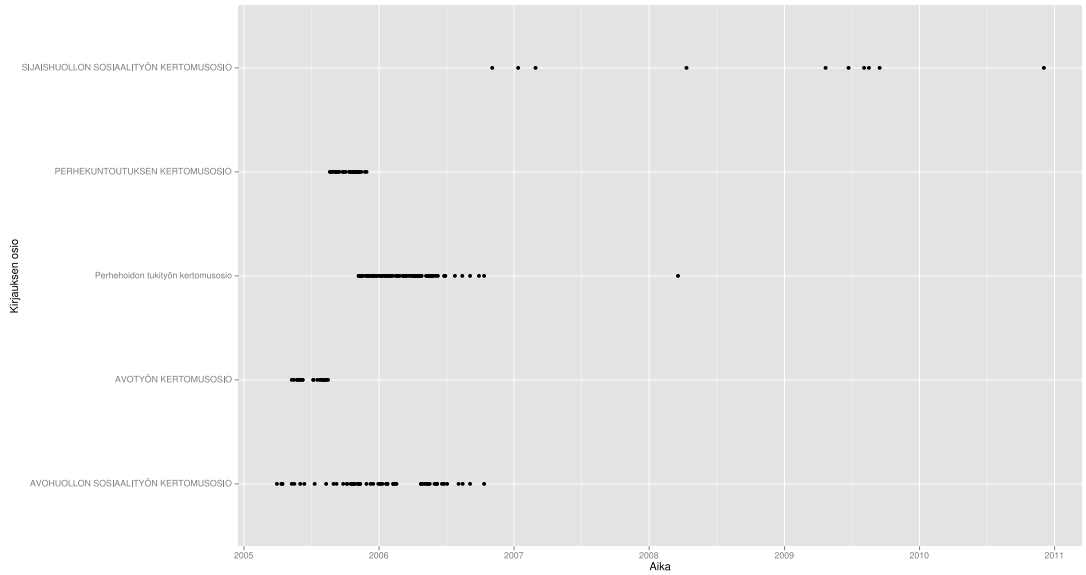
5.1 Visualisointi

Aineistosta tuotettujen visualisointien tarkoituksena on helpottaa yleiskuvan saamista aineistosta. Lisäksi tarkoituksena on luoda aineistoon katsauksia, joissa vain osa sen sisältämästä informaatiosta nostetaan esille.

5.1.1 Kirjausaikajana

Aineiston kirjaukset jakautuvat eri osioihin, jotka liittyvät lastensuojelun eri osa-alueisiin. Kirjausaikajanalla kirjaukset esitetään kuvassa 5.1 esitetyllä tavalla. Kuvan vaaka-akselilta nähdään kirjauksen tekoaika ja pystyakselilla taas esitetään kirjauksien osiot jakamalla eri osioihin kuuluvat kirjaukset eri korkeuksille. Tavoitteena on saada eri sosiaalityön muotojen ajallinen jakautuminen paremmin esille. Kuvan 5.1 toisella rivillä esitetyn perhekuntoutuksen kertomusosion kirjaukset käsittelevät tilannetta, jossa tapauksen perhe on ollut perhekuntoutuksessa. Perhekuntoutuksessa perhe sijoitetaan laitosyksikköön ja sijoitus voidaan toteuttaa esimerkiksi lastensuojelulaitoksessa tai päihdehuollon hoitolaitoksessa [THL 2007]. Ensimmäisellä rivillä olevat sijaishuollon sosiaalityön kirjaukset taas kertovat lapsen olevan sijoitettuna kodin ulkopuolelle. Lapsen sijaishuollolla tarkoitetaan *Lastensuojelun käsikirjan* [THL 2007] mukaan: “huostaanotetun, kiireellisesti sijoitetun tai lastensuojelulain 83 §:ssä tarkoitetun väliaikaismääräyksen nojalla sijoitetun lapsen hoidon ja kasvatuksen järjestämistä kodin ulkopuolella. Poikkeuksellisesti huostaanotettu lapsi voidaan sijoittaa myös kotiin”. Loput riveistä kuvaavat muita sosiaalityön muotoja. Esitetyn, varsin yksinkertaisen kuvaajan perusteella saadaan nopeasti kuva asiakkuuteen liittyvistä sosiaalityön muodoista ja mahdollisten sijoitusten ja laitosjaksojen ajallisesta jakautumisesta. Lisäksi kuvaajalta saadaan samalla kuva asiakassuhteen tiiviydestä. Pitkä aikaväli kirjausten välillä nostaa esille kysymyksen asiakkaan tilanteesta. Kirjausten puuttuminen saattaa

Kuva 5.1 Kirjausaikajana



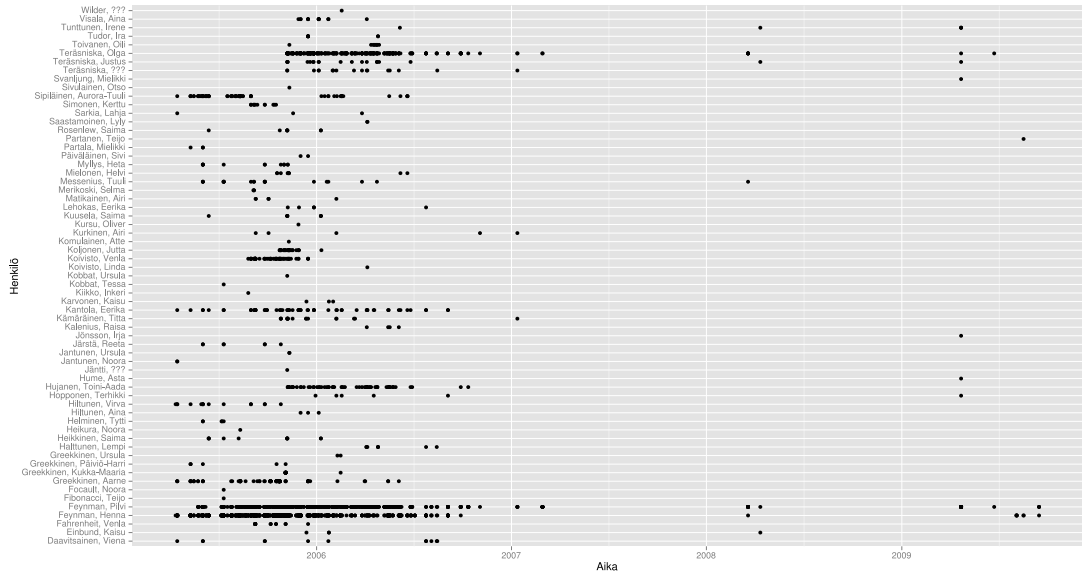
johtua joko perheen onnellisesta tilanteesta tai sosiaalityöntekijän vaihtumisesta tai resurssipulasta. Kirjausaikajana joka tapauksessa helpottaa edellä mainittujen tilanteiden havaitsemista antamalla ajallisen kuvauksen tapauksen kirjausten jakautumisesta.

5.1.2 Henkilöaikajana

Esitystavaltaan henkilöaikajana noudattaa samaa kaavaa kuin kirjausaikajana. Erona on, että henkilön esiintymät kirjauksissa ovat pystyakselilla kirjauksien osiotietojen sijaan. Kuvassa 5.2 on esitetty henkilöaikajana. Voimakkaimmin aikajanalta erottuvat henkilöt ovat Pilvi ja Henna Feynman, Eerika Kantola ja Olga Teräsniska. Näistä Pilvi ja Henna Feynman ovat tapauksen lapsi ja äiti, Kantola taas eräs sosiaalityöntekijöistä. Olga Teräsniskan rooli tapauksessa on sijaisperheen äiti.

Tällaisenaan kuva 5.2 antaa varsin karkean kuvan tapauksen henkilöistä. Toisaalta tapaukseen liittyvien henkilöiden kokonaismäärästä ja esiintymistä on vaikeaa saada yhtä helposti yleiskuvaa pelkästään kirjauksia lukemalla. Kuvaa voitaisiin edelleen parannella karsimalla esitettyjen henkilöiden määrää sekä lisäämällä näille tieto heidän roolistaan tapauksessa. Roolien tunnistamista on kuvailtu tarkemmin osiossa 5.2.1. Nykyisessä kuvassa on esitetty kaikki tunnistetut henkilöt, mukaan lukien ne jotka on mainittu vain kerran. Tällaiset kerran mainitut henkilöt voivat tapauksen kannalta olla olennaisia, esimerkkinä yksittäisen lastensuojeluilmoituksen tehnyt henkilö. Toisaalta kerran mainittujen henkilöiden joukkoon mahtuu myös vain yhdessä asiakastapaamisessa paikalla ollut sosiaalityön opiskelija, joka ei ole tapauksen asiasisällön kannal-

Kuva 5.2 Henkilöaikajana



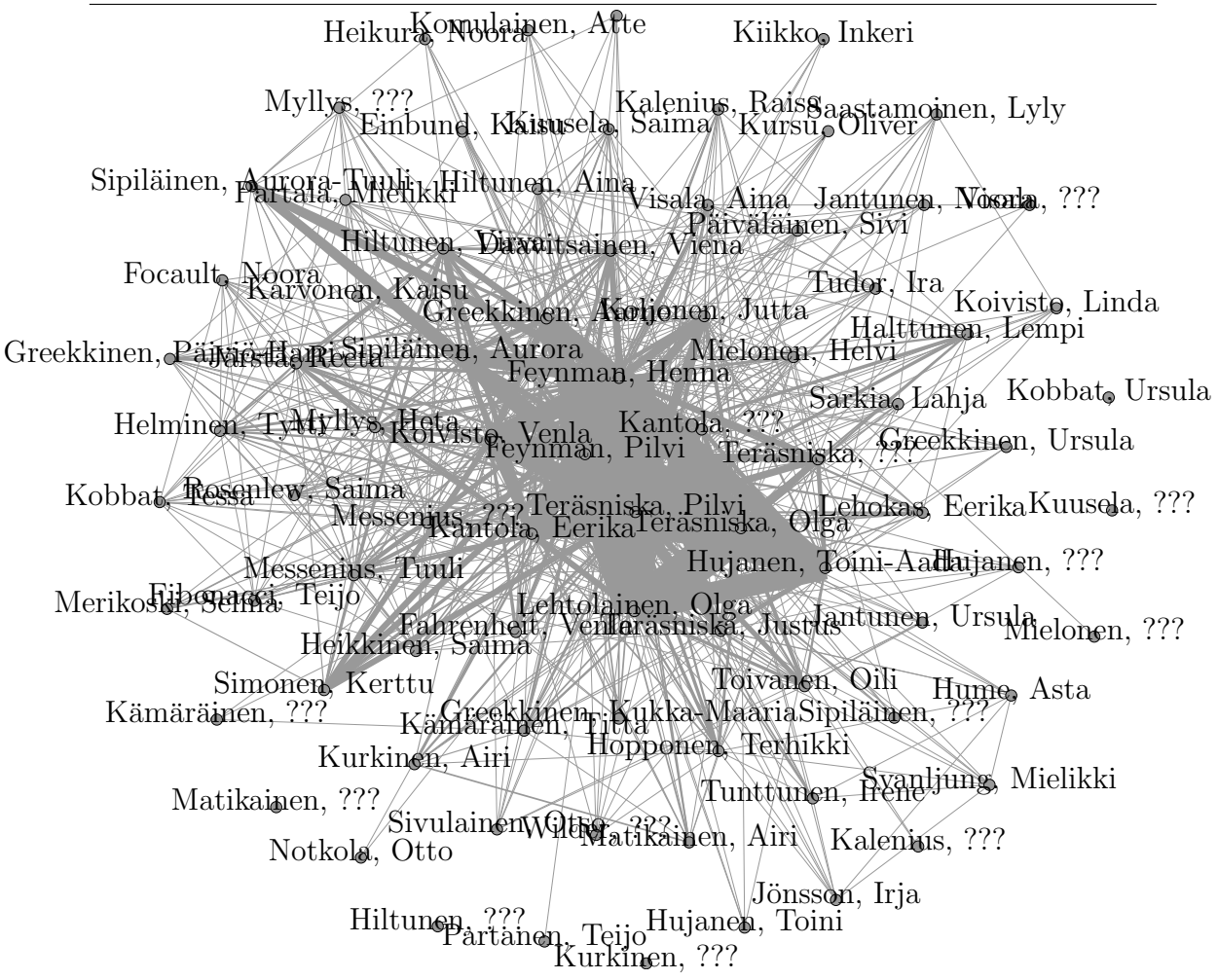
ta relevantti. Kuvaa voitaisiin lisäksi selkeyttää ryhmittelemällä henkilöt eri ryhmiin. Mahdollisia henkilöiden rooleihin perustuvia ryhmiä esitellään osiossa 5.2.1.

5.1.3 Henkilöesiintymägraafi

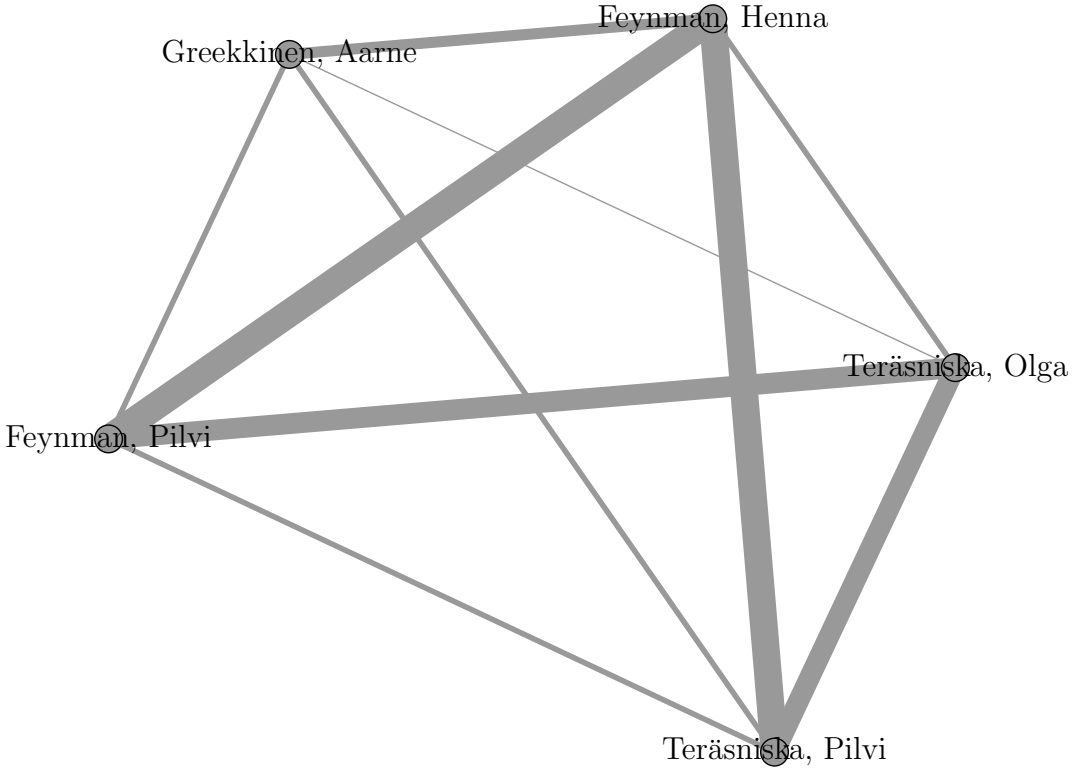
Henkilöaikajanaa voidaan pitää henkilöesiintymägraafin yksinkertaistettuna staattisena aikajanaesitysmuotona, sillä ne molemmat perustuvat lähtökohtaisesti samaan dataan. Henkilöesiintymägraafiin on tallennettu tieto tapaukseen liittyvästä sosiaalisesta graafista, toisin sanoen henkilöistä ja heidän yhteisesiintymistään. Kahden henkilön välillä on yhteisesiintymä, mikäli heidät mainitaan samassa kirjauksessa. Visuaalisesti henkilöt ovat graafin solmuja ja kaaret yhdistävät samassa tapauksessa esiintyviä henkilöitä. Henkilöidenesiintymien lisäksi graafiin on tallennuttu kunkin henkilön esiintymien lukumäärä sekä ensimmäisen ja viimeisen maininnan aikaleimat. Näiden lisätietojen hyödyntämistä on havainnollistettu kuvissa 5.3 ja 5.4. Ensimmäisessä kuvassa on tapauksen koko henkilöesiintymägraafi, jossa useimmin mainitut henkilöt ovat keskemmällä graafia ja kaarien paksuus ilmaisee yhteisesiintymien lukumäärää. Graafi on sellaisenaan varsin käyttökelvoton, mutta sen karsittu versio kuvassa 5.4 parantaa tilannetta. Karsitussa graafissa on otettu mukaan vain henkilöt, jotka mainitaan kirjauksissa vähintään sata kertaa. Laajemman graafin mukaisesti kaarien paksuudet ilmaisevat yhteisesiintymien määrän.

Sekä solmuihin, että kaariin on liitetty tieto niiden esiintymisajasta. Tällöin Gephi-graafieditorin [Bastian, Heymann ja Jacomy 2009] avustuksella sosiaalinen verkosto

Kuva 5.3 Henkilöesiintymägraafi ilman karsintaa.



Kuva 5.4 Kuvasta 5.3 suodatettu henkilöesiintymägraafi, jossa esiintyvät vain vähintään sata kertaa tapauksessa mainitut henkilöt.

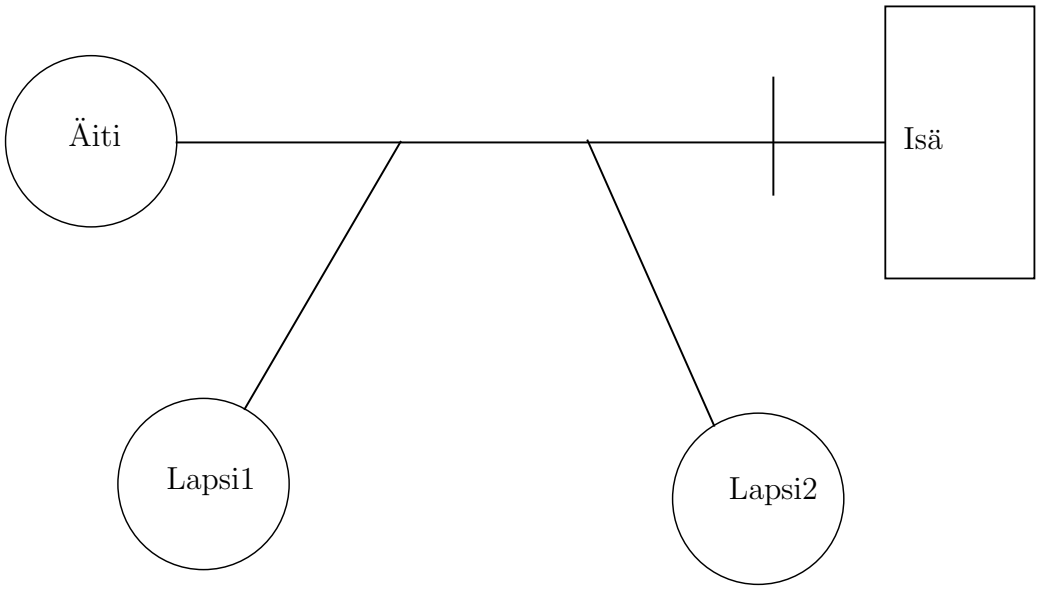


voidaan nähdä animaationa tai verkoston tilaa voidaan havainnoida tietyn aikaikkunan puitteissa. Solmuihin voidaan lisätä myös muuta informaatiota, jonka perusteella graafia voidaan suodattaa reaaliaikaisesti. Graafista voidaan rajautua tarkastelemaan vain niitä osia, jotka liittyvät tiettyyn ajanjaksoon, samoin rajaus voidaan tehdä esimerkiksi asettamalla alaraja sille, kuinka useassa kirjauksessa henkilön on esiinnyttävä. Näin voidaan keskittyä vain tapauksen kannalta tärkeimpiin henkilöihin. Edellä kuvattujen rajausten avulla voitaisiin vastata esimerkiksi kysymykseen: ketkä olivat tapauksen kannalta tärkeitä henkilöitä asiakkuuden ensimmäisen vuoden aikana, kun oletetaan tärkeiden henkilöiden olevan sellaisia, jotka mainitaan usein? Kuva 5.4 voisi toimia alustavana vastauksena edeltävälle kysymykselle. Kuvan valikoituneet henkilöt ovat lapsen äiti, sijaisäiti ja äidin poikaystävä sekä tapauksen varsinainen lapsi. Kuten kuvasta huomataan, esiintyy lapsi siinä kahdella eri sukunimellä. Tapauksen alkupään kirjauksissa lapsi esiintyy äitinsä sukunimellä ja loppuvaiheessa sijaisperheen sukunimellä. Nimenvaihdoksia tapahtuu tapauksissa ajoittain, mutta niitä ei ole erikseen dokumentoitu kirjauksiin.

5.1.4 Sosiaalinen verkosto

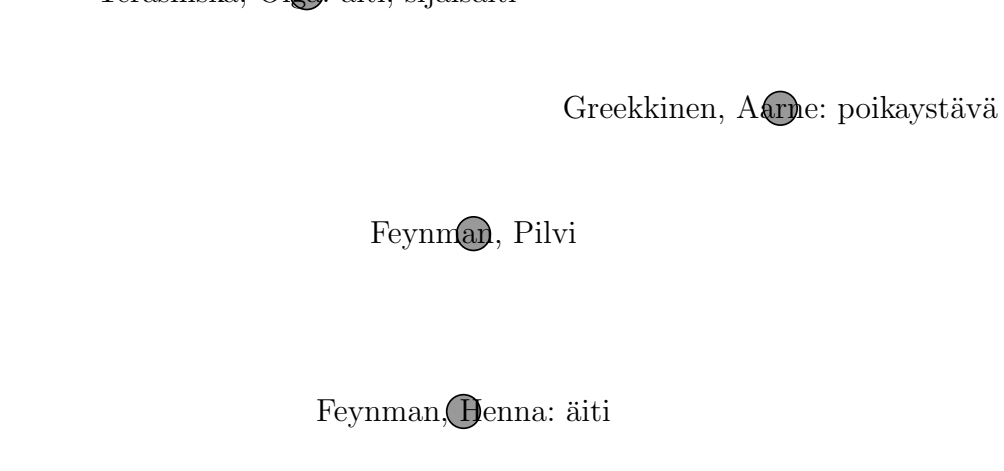
Sosiaalinen verkosto -graafi on henkilöesiintymägraafin jalostettu versio, jossa kuvataan asiakkaaseen liittyvä sosiaalinen verkosto. Käytännössä verkoston voitaisiin ajatella tyypillisesti kattavan perheen, lähiomaiset ja muut perheen tukena olevat läheiset. Taustana tälle visualisoinnille on *Lastensuojelutyön dokumentointi* [Kääriäinen, Leinonen ja Metsäranta 2006] kirjassa oleva sosiaalista graafia kuvaava kaavio. Kuvassa 5.5 on esitetty alkuperäisen kuvan mukailtu versio. Alkuperäinen lähde ei tarkemmin kuvaile kuvassa käytettyä symboliikkaa, mutta voidaan olettaa, että perheen lapset asuvat äitinsä luona ja vanhempien olevan eronneita.

Kuva 5.5 Mukailtu versio ”Lastensuojelutyön dokumentointi”[Kääriäinen, Leinonen ja Metsäranta 2006] kirjassa esitetystä kaaviosta.



Graafin koostamiseen käytetään kohdan 5.1.3 henkilöesiintymägraafia ja kohdan 5.2.1 tunnistettuja henkilöiden rooleja. Tunnistettujen roolien joukosta valitaan sosiaaliseen verkostoon valittavat roolit ja samalla myös rooleihin liittyvät henkilöt. Käytännössä sosiaaliseen verkostoon päätyvät roolit on poimittu kaikkien löydettyjen roolien listalta ja lisätty omalle hyväksyttyjen roolien listalle. Jokaiselle sopivan roolin omaavalle henkilölle lisätään graafiin oma solmu ja solmu nimikoidaan roolilla ja henkilön nimellä. Solmuun voidaan liittää myös aikaväli, jona solmu on osa graafia. Sosiaaliseen verkostoon voidaan siis tarvittaessa liittää myös ajallinen ulottuvuus. Ajallinen tarkastelu on kuitenkin täysin valinnaista ja aikaulottuvuus voidaan myös sivuuttaa. Kuvassa 5.6 on esimerkki aineiston pohjalta päätellystä sosiaalisesta verkostosta. Nykyisessä toteutuksessa solmujen eli henkilöiden välille ei alkuperäisen kuvan vastaisesti

Kuva 5.6 Ehdotelma tapauksen sosiaaliseksi verkostoksi



lisätä kaaria, koska kerätyn aineiston perusteella perheen sisäisiä suhteita ei voida automaattisesti päätellä. Nykyisellään kuvan voidaan kuitenkin todeta olevan alustava ehdotelma tapauksen sosiaalisesta graafista, jota työntekijä voi tarpeen mukaan täydentää ja laajentaa. Esimerkkikuvassa 5.6 esiintyvät roolit ovat keskenään päällekkäisiä ja ristiriitaisia. Päällekkäisyyksien syitä tarkastellaan tarkemmin roolien päättelyä käsittelevässä kohdassa 5.2.1.

5.2 Aineistosta eristettävä data

Edellä kuvattujen tietojen lisäksi aineistosta erotetaan tai voidaan helposti erottaa myös henkilöiden rooleja tai heidän toimintaansa. Nämä eristettävät tiedot tiivistävät sellaisenaankin tapausta, mutta suurempaa hyötyä niistä saadaan jatkokäsittelemällä ja liittämällä ne muun datan mukana osaksi erilaisia yhteenvetoja.

Rooli kuvaa tapaa, jolla henkilö liittyy tapaukseen, esimerkiksi sosiaalityöntekijä tai äiti. Liittämällä roolitiedot mukaan henkilöaikajanaan saadaan janan hyödyllisyyttä kasvatettua huomattavasti. Roolit tuntemalla jana voitaisiin esimerkiksi koostaa sisältämään pelkät työntekijät, jolloin nähtäisiin helpommin keneltä tapausta käsitelleeltä työntekijältä asioita kannattaa kysyä. Samalta janalta voitaisiin myös helposti nähdä tapauksen työntekijöiden vaihtuvuus.

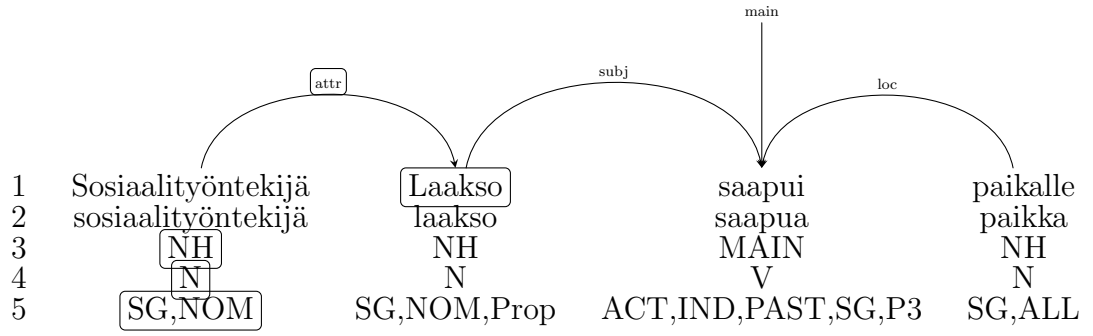
Tapauksen henkilöiden toimintaa voidaan suppeasti tarkastella keräämällä lauseet, joissa henkilö on jäsennyspuun mukaan subjektina. Näin päästään lähemmäs henkilöiden aitoa toimintaa omassa elämässään. Voidaan ajatella, että henkilöaikajanat ja henkilöiden roolit ovat tietoa, joka liittyy todelliseen henkilöön olla-verbin kautta. Lauseiden jäsennyspuita tarkastelemalla taas nähdään, mitä henkilöt tekevät, pelkän maininnan kohteena tai roolissa olemisen lisäksi. Sama tarkastelu voidaan laajentaa myös

esimerkiksi aineiston henkilöiden kertomiin asioihin.

5.2.1 Aineiston henkilöiden roolit

Henkilöaikajanalla esiintyville henkilöille voidaan päätellä heitä kuvaavat roolit. Roolien karkean jaon esimerkkinä tapauksissa esiintyvät henkilöt voidaan aineiston perusteella jakaa tapauksen asiakkaan sosiaaliseen verkostoon, työntekijöihin, asiantuntijoihin ja muihin henkilöihin. Tapauksen asiakkaan sosiaaliseen verkostoon voidaan laskea kuuluviksi esimerkiksi perhe, suku ja tuttavat. Työntekijöihin kuuluvat esimerkiksi tapausta hoitava sosiaalityöntekijä ja muu vastaava henkilökunta. Asiantuntijoita ovat esimerkiksi tapauksissa toisinaan esiintyvät lääkärit ja psykiatrit. Tapauksesta riippuen myös esimerkiksi opettajat tai muut työnsä puolesta tapaukseen liittyvät henkilöt voitaisiin laskea asiantuntijoiksi. Muihin henkilöihin voitaisiin luokitella tapauksissa satunnaisesti esiintyvät henkilöt. Esimerkiksi asiakaskertomuksissa vain kerran esiintyvät sosiaalialan opiskelijat tai muut satunnaisesti mainitut henkilöt voitaisiin luokitella tähän kaatoluokkaan. Taulukossa 5.1 on esitetty eräästä aineiston tapauksesta pääteltyjä rooleja. Nykyisellään roolien tunnistaminen voisi toimia osana puoliautomaattista roolien päättelyä, jossa lähtökohtana toimivat roolit tuotettaisiin automaattisesti ja käyttäjä tarkastaisi ja korjaisi ehdotuksen lopulliseen muotoon.

Kuva 5.7 Roolien tunnistamista havainnollistava jäsennyspuu.



Roolit on eristetty jäsennetystä aineistosta sanojen riippuvuuksia käyttämällä. Nimisanojen attribuuteiksi (attr) jäsennetyt perusmuotoiset (SG) substantiivilausekkeen (NH) pääsanoiksi tunnistetut substantiivit (N) on valittu roolikandidaateiksi. Löydetyt kandidaatit on käyty käsin läpi ja niiden perusteella on muodostettu niin sanottu valkoinen lista (white list) rooleja kuvaavista sanoista. Lopulliset tunnistetut roolit on karsittu käyttäen tätä listaa. Kuvassa 5.7 on ympyröimällä korostettu roolien tunnistamisen kannalta relevantteja kohtia jäsennyspuussa. Jäsennyspuun merkinnät tulkitaan kuten kuvassa 3.11. Aineiston RDF-tallennusmuotoa ja sen käsittelyä varten toteutet-

Taulukko 5.1: Henkilöitä ja näille tunnistettuja rooleja.

Nimi	Rooli
Fahrenheit, Venla	sosiaalityöntekijä
Feynman, Henna	äiti
Greekkinen, Aarne	poikaystävä
Greekkinen, Kukka-Maaria	äiti
Greekkinen, Ursula	täti
Hiltunen, Virva	terveydenhoitaja
Jäntti	sosiaalityöntekijä
Järstä, Reeta	sosiaalityöntekijä
Kantola	sosiaalityöntekijä
Kantola, Eerika	sosiaalityöntekijä,öntekijä
Koivisto, Venla	työntekijä
Lehokas, Eerika	työntekijä
Messenius,	tohtori,lääkäri
Messenius, Tuuli	lääkäri
Mielonen, Helvi	sosiaalityöntekijä
Myllys, Heta	psykiatri,lääkäri
Partala, Mielikki	opiskelija
Sipiläinen, Aurora	avotyöntekijä
Teräsniska, Olga	äiti,sijaisäiti
Toivanen, Oili	psykiatri

tua kirjastoa käyttäen voidaan roolikandidaattien tunnistaminen tulkita graafin täsmäys -tehtäväksi. Roolien tunnistamisessa on tämän tulkinnan mukaan kyse kahden toisistaan halutulla tavalla riippuvan sanasolmun etsimisestä siten, että solmut samalla täyttävät niille asetetut kieliopilliset ehdot.

Kuvattu roolien tunnistamismenetelmä hyödyntää jäsennystulosta vain suppeasti. Nykyiselleen rooleja varten on otettu huomioon vain ensimmäinen nimestä riippuva sana, joten kaikki tunnistetut roolit ovat yksisanaisia. Parempi tulos saataisiin hyödyntämällä jäsentimen tuottamaa jäsennystä myös muiden sanojen osalta. Taulukossa 5.1 esitettyjä löydettyjä rooleja tarkastelemalla havaitaan kuitenkin, kuinka jo varsin yksinkertaisilla menetelmillä voidaan tekstistä löytää mielekkäitä rooleja.

Kuten taulukosta 5.1 huomataan, ei edellä kuvattu yksinkertainen menetelmä ole täydellinen. Esimerkkinä yksinkertaisen menetelmän ongelmista on usean eri henkilön luokitteleminen äidiksi. Mikään näistä luokitteluista ei kuitenkaan ole täysin väärin, vaikkei taulukon perusteella olekaan täysin selvää, kuka henkilöistä on äiti, sijaisäiti ja isoäiti. Kaikilla äidiksi tunnistetuilla henkilöillä on kuitenkin tosiasiassa jokin edellä mainituista rooleista. Toinen taulukosta havaittava ongelma on roolien vaillinainen

tunnistaminen, joka johtuu roolien nykyisestä rajautumisesta vain yksittäiseen sanaan. Taulukossa esiintyvät *työntekijä*-roolit ilmentävät tätä nykyisen lähestymistavan puutetta. Työntekijä ei sellaisenaan ole mielekäs rooli, joten se tarvitsisi *työntekijä*-sanaa määrittävän sanan ollakseen kokonainen rooli-ilmaus.

5.2.2 Aineiston henkilöiden toiminta

Dependenssijäsennin tuottaa sanojen riippuvuudet suhteessa toisiinsa. Näiden riippuvuuksien joukossa ovat muun muassa verbeihin liittyvät subjekti- ja objektiriippuvuudet. Jäsennettyjä lauseita läpikäymällä voidaan siis saada selville, mitä tapauksessa esiintyvät henkilöt tekevät. Nämä tiedot voidaan yhdistää aineistosta tunnistettuihin nimisanoihin ja näihin liittyviin henkilöihin, jolloin saadaan roolien lisäksi selville jotain henkilöiden toiminnasta. Ongelmana on, ettei kaikissa lauseissa henkilöitä mainita nimeltä, vaan heihin tyydytään viittaamaan pelkällä persoonapronominilla. Tähän viittausongelmaan ei tässä työssä perehdytä, koska se muodostaa oman tutkimusalueensa, joka tunnetaan nimellä anaforisten viittausten päättely (anaphora resolution). Katsauksen aiheeseen esittävät esimerkiksi Poesio, Ponzetto ja Versley [Poesio, Ponzetto ja Versley 2011].

Henkilöiden toiminnan selvittämistä lähestytään samankaltaisena graafin täsmäys -ongelmana kuin henkilöiden roolien päättelyä kohdassa 5.2.1. Nykyisessä toteutuksessa etsitään lauseita, joissa jokin nimisana on verbin subjektina. Täsmätyistä lauseista kerätään tämän jälkeen rekursiivisesti kaikki verbistä riippuvat sanat. Nämä riippuvat sanat ilmaisevat muun muassa tekemisen aikaa, paikkaa, kestoa ja tapaa. Kuten roolien tunnistamisessakin, on nykyinen menetelmä varsin karkea ja sen voidaankin todeta olevan liian yksinkertainen malli lauserakenteesta käytetyn aineiston kohdalla.

Alla on esitelty muutamia lauseita, joista on löydetty haluttu lauserakenne. Esimerkkilauseet on esitetty korostettuna ja niiden alle on taulukoitu tunnistetusta nimisanasta koostuva subjekti ja verbi, sekä verbistä riippuvat sanat. Riippuvat sanat on esimerkeissä erotettu pilkuilla toisistaan ja järjestetty alkuperäisen lauseen mukaiseen järjestykseen.

`Perhetyöntekijä kertoo, että Amalia oli ollut sekava vapun jälkeisenä maanantaina.`

Subjekti	Predikaatti	Predikaatin määritteet
Amalia	oli	ollut, vapun, jälkeisenä, maanantaina

Amalia kertoi Karrin olevan vielä hänelle loukkaantunut, koska joutunut lähtemään toiseen paikkaan.

Subjekti	Predikaatti	Predikaatin määritteet
Karrin	olevan	vielä, loukkaantunut

Hänelle sopii, että Helmi vie pojat Särkänniemeen kello 12.00 ja Viola voi halutessaan viedä pojat Suojakodille syömään josta Helmi hakee heidät.

Subjekti	Predikaatti	Predikaatin määritteet
Helmi	vie	pojat, Särkänniemeen, ja, Viola, voi, halutessaan, viedä, pojat, Suojakodille

Lääkäri kertoo, että Amalia on aika yksinäinen, ja tarvitsisi normaalia arkea ja ystäviä joiden kanssa jakaa asioita.

Subjekti	Predikaatti	Predikaatin määritteet
Amalia	on	aika, yksinäinen, ja, tarvitsisi, normaalia, arkea, ja, ystäviä

Impi kertoi, että lääkärin tutkimusten perusteella Karri on terve.

Subjekti	Predikaatti	Predikaatin määritteet
Karri	on	terve

Jatkossa tärkeää, että Amalialla on säännöllinen hoitokontakti.

Subjekti	Predikaatti	Predikaatin määritteet
Amalialla	on	säännöllinen, hoitokontakti

Kuten esimerkkilauseista havaitaan, tavoittaa käytetty lausekehahmo vain osan lauseiden informaatiosta. Korostuneesti jää puuttumaan tieto siitä, kuka lauseessa on ollut kertojana.

5.2.3 Aineiston henkilöiden kertomat asiat

Aineistolle on tyypillistä sisältää paljon lauseita, joissa joku kertoo jotakin. *Kertoa* onkin aineiston yleisin verbi *olla*-verbin jälkeen. Yleisti voidaankin luonnehtia sosiaalityöntekijöiden kuvailevan tapahtumia ja ihmisiä sekä kirjaavan toisten kertomia havaintoja. Koska tämäntyyppinen kerrontatapa eroaa tavanomaisesta, suoraviivaisemmasta proosatekstistä, vaadittaisiin sen informaatiosisällön tavoittamiseen aineiston kertomispainotteisen rakenteen paremmin huomioon ottavia täsmäytettäviä lauserakenteita. Tässä osiossa kuvattu tiedon eristäminen lauseista kärsii roolien tunnistamista enemmän aineiston jäsennyksessä tapahtuvista virheistä ja puutteista. Miltei kaikkien esimerkkilauseiden jäsentäminen on vajavaista eli kaikkien sanojen riippuvuuksia suhteessa toisiinsa ei pystytä päättelemään. Useimmiten kyseessä ovat kuitenkin vain yksittäiset sanat. Jäsennykseen liittyvien ongelmien voidaan epäillä olevan peräisin muun muassa aineistossa käytetyistä upotetuista lauserakenteista, joissa lause esiintyy toisen lauseen alisteisena jäsenenä [Karlsson 2008]. Aiemmista esimerkkilauseista tätä rakennetta havainnollistaa lause: “Impi kertoi, että lääkärin tutkimusten perusteella Karri on terve”.

Alla on listattu lauseita, joissa esiintyy *kertoa*-verbi jossakin muodossa. Kunkin lauseen alla on esitetty *kertoa*-verbistä riippuvat sanat. Kuten edellisissäkin esimerkeissä, on riippuvaiset sanat esitetty esiintymisjärjestyksessään.

- `Sosiaalityöntekijä kertoi seuraavaa viestiä hoitotiimille.`
  - Sosiaalityöntekijä, kertoi, seuraavaa, viestiä, hoitotiimille
- `Taija kertoi, että Joosepin lääkitystä on lisätty ja se on rauhoittanut hieman perheen elämää.`
  - Taija, kertoi, että, lääkitystä, on, lisätty, ja, se, on, rauhoittanut, hieman, perheen, elämää
- `Soitto Taija Meskaselle, kertoo hakeneensa sairaslomaa täksi viikoksi ja on nyt XXX kriisiryhmässä.`
  - kertoo, hakeneensa, sairaslomaa, täksi, viikoksi, ja, on, nyt, kriisiryhmässä
- `Äiti kertoi psyykkisen vointinsa olevan todella huono ja työnteon olevan vaikeaa tällä hetkellä.`
  - Äiti, kertoi, vointinsa

Vaikka esimerkkinä olevat kertomiseen liittyvät sanajoukot ovatkin vajavaisia, tavoittaa yksinkertainen lausekehahmo tässäkin edellisen osion tapaan olennaisia seikkoja lauseen kuvaamasta kerronnasta.

5.3 Aineiston muuttaminen hypertekstiksi

Visualisointien lisäksi aineiston lähestyttävyyttä parannetaan muuntamalla se hypertekstiksi. Tätä tarkoitusta varten aineiston esittämiseen käytetään Semantic MediaWiki -ohjelmistoa [Krötzsch et al. 2007]. Semantic MediaWiki, lyhemmin SMW, on laajennus Wikipedian käyttämään MediaWiki-ohjelmistoon ja se mahdollistaa tekstin semanttisen annotoinnin. Lyhyesti todettuna SMW mahdollistaa ominaisuuksien lisäämisen sivuille. Ominaisuudet noudattavat RDF:n tapaan subjekti-predikaatti-objekti -kolmikko jaottelua. Tavanomaisesti subjekti on kyseinen sivu ja predikaatti kuvaa ominaisuuden nimen. Ominaisuuden arvo voi olla esimerkiksi luku tai sivu. Esimerkiksi Suomea käsittelevään sivuun voidaan liittää predikaatti 'väkiluku', jonka arvona on kokonaisluku. Samaan sivuun voitaisiin liittää myös predikaatti 'pääkaupunki', jonka arvona on Helsingistä kertova sivu. SMW:n avulla linkkejä voidaan siis tyypittää, jolloin sivujen välisiä suhteita voidaan kuvata niin, että semanttisia hakuja voidaan niiden perusteella tehdä.

5.3.1 Nykyinen toteutus ja sen mahdolliset laajennukset

Hypertekstimuodossa jokainen kirjaus esitetään omalla sivullaan, kuten myös jokainen henkilö. Kirjauksessa mainitut henkilöt linkitetään henkilöstä kertovaan sivuun. Linkit myös tyypitetään, jotta henkilöitä voidaan myöhemmin hakea kirjauksista. Lisäksi tapauksen aikaleima ja osio esitetään SMW:n ymmärtämässä muodossa, jolloin niihin voidaan kohdistaa hakuja. Kuvassa 5.8 on eräs Semantic MediaWiki:n tallennettu kirjaussivu.

Edellä esitettyjä annotaatioita hyödyntämällä voidaan aineistosta hakea esimerkiksi kirjauksia, joissa esiintyvät halutut henkilöt ja jotka on tehty tiettynä ajanjaksona. SMW:n avulla voidaan myös muodostaa yhteenvetoja predikaateista ja niiden arvoista. Yhteenveto voi koskea esimerkiksi tiettyä henkilöä siten, että henkilösivulla on automaattisesti luotu lista ja linkki kaikkiin kirjauksiin, joissa henkilö mainitaan. Kuvassa 5.9 on Semantic MediaWiki:n tuottama listaus kirjauksissa esiintyvistä henkilöistä ja kuvassa 5.10 ovat henkilön maininnat eri kirjauksissa.

Henkilöstä kertovalle sivulle on myös mahdollista lisätä automaattisesti tieto henkilön roolista tapauksessa hyödyntämällä kohdassa 5.2.1 suoritettua roolien päättelyä. Roolitieto voitaisiin samalla linkittää siihen kirjaukseen, jonka perusteella rooli on päätelty. Näin toimimalla tehtäisiin käyttäjälle näkyväksi kirjaus, johon roolitiedon päättely perustuu. Samalla käyttäjälle tarjoutuu myös mahdollisuus tarkastaa päätellyn roolin oikeellisuus.

Lisäksi SMW tarjoaa hakutoiminnallisuuden ja mahdollisuuden metadatan lisäämiseen hypertekstimuotoisiin kirjauksiin. Aiemmin mainittujen tietojen lisäksi kirjauksiin voitaisiin lisätä annotaatioita, jotka kertoisivat esimerkiksi kirjauksessa käsitellyistä ai-

Kuva 5.8 Kirjaussivu Semantic MediaWiki:n esittämänä.

Tapaus 1: kirjaus 229

Kirjausaika: 2006-5-25
Kirjaaja: Tapaus 1: Hujanen, Toini
Kirjausosio: Perhehoidon tukityön kertomusosio

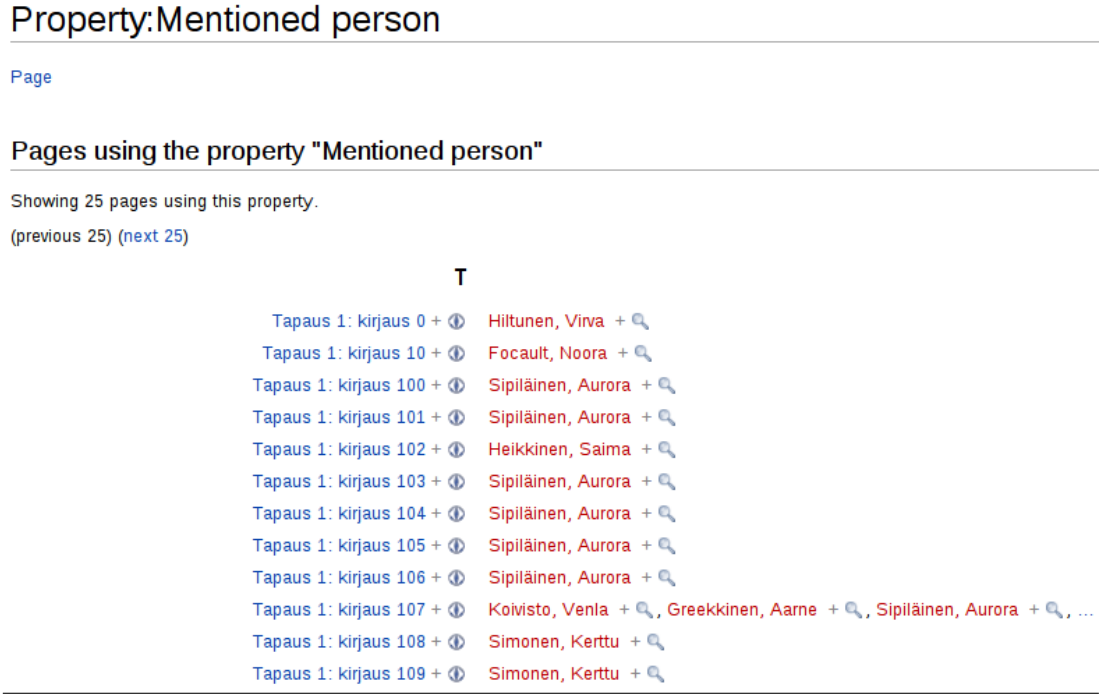
Tekstissä mainitut henkilöt: Hujanen, Toini

Edellinen Seuraava

250506 Toini-Aada Hujanen Perhehoidon Tukiryhmä. Puhelinkeskustelu sijaisäiti Olga Teräsniskan kanssa. Olga kertoi, että Pilvin olo alkoi eilen illalla ko tyytyväisenä lattialla. Tauti oli eilen iskenyt sijaisäiti Olgaan ja hän oli kuumeessa, joten siirrämme huomiseksi aiotun käynnin ensi viikolle, sovitaan siitä seuloista. Hujanen Toini

- 250506 Toini-Aada Hujanen Perhehoidon Tukiryhmä.
- Puhelinkeskustelu sijaisäiti Olga Teräsniskan kanssa.
- Olga kertoi, että Pilvin olo alkoi eilen illalla kohentua ja kuume aleni ja Pilvi oli jo yön nukkunut omassa sängyssään ja nyt leikki tyytyväisenä lattialla.
- Tauti oli eilen iskenyt sijaisäiti Olgaan ja hän oli kuumeessa, joten siirrämme huomiseksi aiotun käynnin ensi viikolle, sovitaan siitä maanantaina.
- Lupasimme huomenna soitella Pilvin äidille tapaamisista ja seuloista.

Kuva 5.9 Semantic MediaWiki:n yhteenveto kirjauksissa esiintyvistä henkilöistä.



Property:Mentioned person

Page

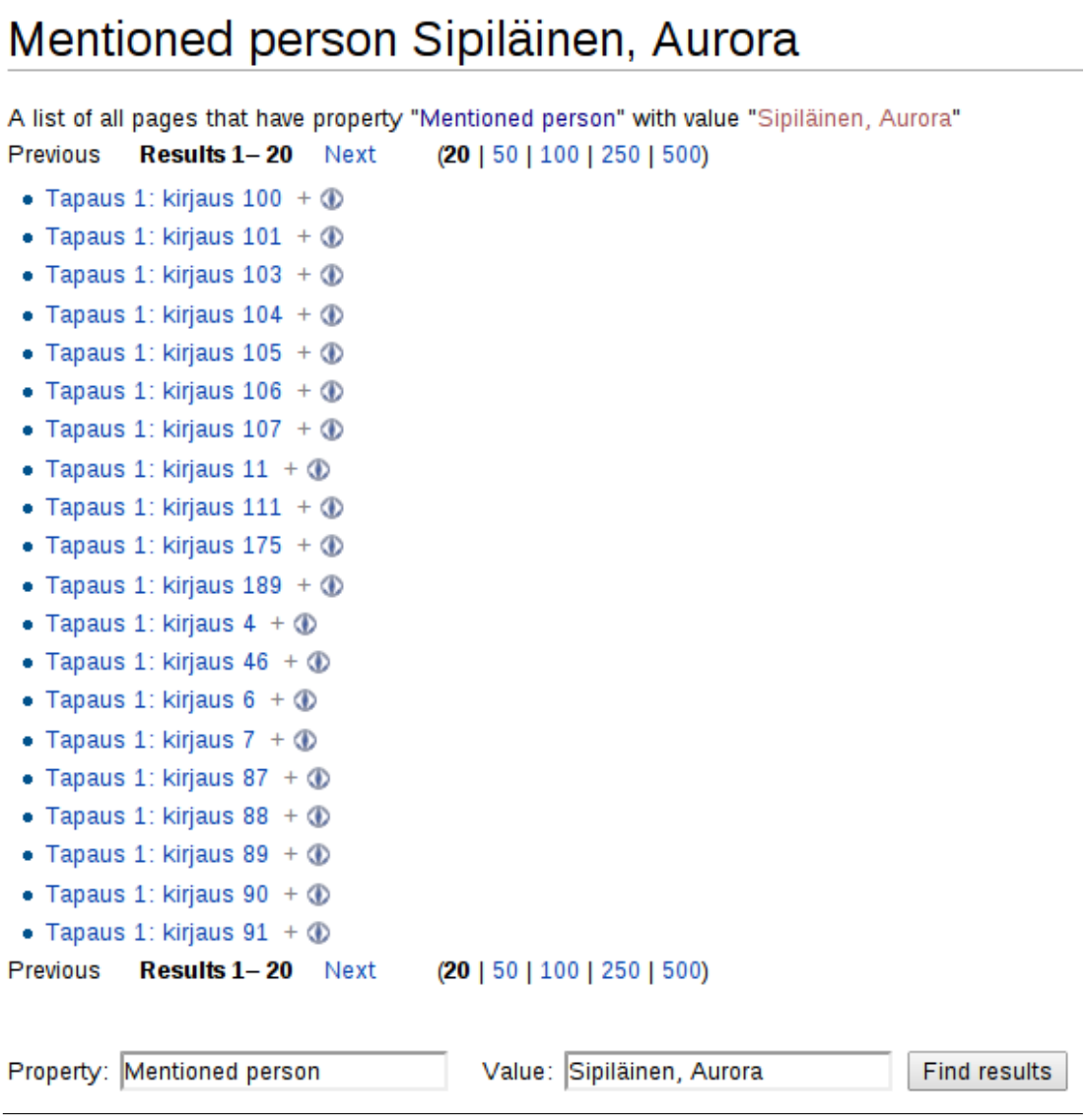
Pages using the property "Mentioned person"

Showing 25 pages using this property.

(previous 25) (next 25)

T	
Tapaus 1: kirjaus 0	Hiltunen, Virva
Tapaus 1: kirjaus 10	Focault, Noora
Tapaus 1: kirjaus 100	Sipiläinen, Aurora
Tapaus 1: kirjaus 101	Sipiläinen, Aurora
Tapaus 1: kirjaus 102	Heikkinen, Saima
Tapaus 1: kirjaus 103	Sipiläinen, Aurora
Tapaus 1: kirjaus 104	Sipiläinen, Aurora
Tapaus 1: kirjaus 105	Sipiläinen, Aurora
Tapaus 1: kirjaus 106	Sipiläinen, Aurora
Tapaus 1: kirjaus 107	Koivisto, Venla, Greekkinen, Aarne, Sipiläinen, Aurora, ...
Tapaus 1: kirjaus 108	Simonen, Kerttu
Tapaus 1: kirjaus 109	Simonen, Kerttu

Kuva 5.10 Semantic MediaWiki:n listaus henkilön esiintymisistä kirjauksissa.



heista. Kirjauksien aiheet voitaisiin päätellä niissä käytetyn sanaston tai avainsanojen perusteella. Mahdollisia aiheita voisivat olla esimerkiksi toimeentulo, kotikäynti, päihteet tai verkostoneuvottelu. Kirjauksissa esiintyviä aiheita voitaisiin käyttää eräänlaisina tapausta kuvaavina tai tiivistävinä asiasanoina. Näille asiasanoille saadaan kirjausten kautta myös ajallinen ulottuvuus, jolloin ne voitaisiin esittää myös aikajanalla. Janan avulla voitaisiin helposti tarkastella esimerkiksi sitä, ovatko päihteidenkäyttö tai toimeentulo-ongelmat tapausta koko ajalta leimaavia seikkoja, vaiko jo ratkaistuja ongelmia. Liittämällä kirjauksiin metadataa kirjauksessa käsitellyistä aiheista voitaisiin tehdä hakuja, jotka rajautuvat tiettyjä aiheita käsitteleviin kirjauksiin. Eräs käyttötarkoitus tällaiselle toiminnallisuudelle olisi päihteisiin liittyvissä tapauksissa läpikäynnin keskittäminen vain osaan kirjauksista.

5.3.2 Hypertekstimuotoisten kirjausten mahdolliset edut

Aineiston muuntamisella hypertekstiksi saadaan osittain helpotettua henkilöihin liittyvien tiedontarpeiden täyttämistä. Edellä kuvattu aineiston muuntaminen hypertekstiksi, metadatan sekä linkkien lisääminen on toteutettu automaattisesti ilman käyttäjän väliintuloa. Tämä prosessointi voitaisiin toteuttaa jo kirjausvaiheessa osana asiakastietojärjestelmää. Tällöin käyttäjän kirjoitettua kirjauksen kävisi järjestelmä sen läpi tunnistaen henkilöt, sekä tehden muut halutut annotaatiot. Tämän jälkeen järjestelmä esittäisi kirjaajalle tunnistamansa ja päättelemänsä henkilöt, sekä tehdyt annotaatiot. Mikäli kirjaaja toteaisi päätelmät oikein tehdyiksi, lisäisi järjestelmä hyväksytyt linkit ja annotaatiot automaattisesti tekstiin. Väärin päätellyt annotaatiot kirjaaja voisi ennen hyväksymistä hylätä tai korjata. Tässä toimintatavassa järjestelmä tekisi varsinaisen työn ja käyttäjälle jäisi vain tulosten tarkastaminen ja mahdollinen korjaaminen.

Edellä kuvattu työnkulku on suunniteltu soveltumaan mahdollisimman luontevaksi osaksi sosiaalityöntekijöiden nykyistä työtapaa. Rakennetta ja koneluettavuutta pyritään lisäämään kirjauksiin muuttamatta itse kirjauksien kirjoittamisprosessia. Itse kirjauksen kirjoittaminen tapahtuisi siis täysin samalla tavalla kuin tähänkin asti, vapaata tekstiä käyttäen. Lähestymistapaa voikin kuvata “alhaalta ylös” suuntautuvaksi. Kirjaukset otetaan annettuina sellaisina kuin ne ovat ja niihin vain lisätään annotaatioita. Vastakohtana tälle tavalle toimii rakenteinen kirjaaminen, jossa kirjaukset pilkotaan järjestelmän valmiina antamien otsikoiden alle. Otsikot muodostavat hierarkian, jossa esimerkiksi potilaskertomuksissa annettu lääkitys kirjataan sanallisesti otsikkohierarkian “Toteutus/Lääkehoito” alle. [Kuurne 2009] Rakenteinen kirjaaminen vaatii siis muutoksia itse kirjauksen kirjoittamiseen määrittelemällä kirjauksiin valmiit otsikot ja rakenteen. Rakenteisen kirjaamisen lähestymistapaa voidaankin kuvata “ylhäältä alas” suuntautuvaksi, jolle tässä työssä esitetty olemassa olevia työtapoja lähtökohtana käyttävä malli tarjoaa vaihtoehdon.

6 Toteutetut ja käytetyt työkalut

Luonnollisen kielen käsittelyyn on tarjolla melko paljon erilaisia työkaluja ja kieliresursseja. Niiden määrä on myöskin koko ajan kasvussa, etenkin suomen kielen osalta. Huomattavaa edistystä on tapahtunut jo pelkästään tämän työn tekemisen aikana. Olemassa olevien työkalujen ja resurssien ongelmana on kuitenkin usein niiden sovellettavuus pienten kielialueitten kielille. Tästä johtuen tätä työtä varten on toteutettu Anonymisoivaksi annotaattoriksi, lyhemmin AA, nimetty ohjelmisto aineiston käsittelyä varten. Tämän lisäksi on toteutettu RDF-NLP niminen kirjasto RDF-muotoon tallennetun luonnollisen kielen tekstiaineiston käsittelyä varten. Edellisten lisäksi on toteutettu joukko erilaisia pienempiä ohjelmia esimerkiksi lauseiden etäjäsentämistä, visualisointia sekä tiedostomuotomuunnoksia varten.

6.1 Anonymisoiva annotaattori

Lastensuojelun asiakaskertomusten anonymisointiin ja annotointiin kehitetyn **Anonymisoivan annotaattorin** kehitys on ollut varsin orgaanista. Vaikka sitä tässä työssä käsitellään yhtenä yksikkönä, koostuu se todellisuudessa useista pienemmistä ohjelmista. Alunperin kehitys lähti liikkeelle tarpeesta poistaa automaattisesti henkilötietoja ja anonymisoida asiakaskertomuksia. Tätä varten toteutettiin ohjelma poistamaan tekstiaineistosta henkilötietoja, puhelinnumeroita ja osoitteita. Edellisten lisäksi aineistossa olevat henkilöiden nimet anonymisoidaan johdonmukaisesti toisilla nimillä. Anonymisoinnin lisäksi AA annotoi tekstiä esimerkiksi nimisanojen ja ajanilmausten osalta. Annotointitietoa käytetään myöhemmässä vaiheessa, kun päätellään kaikki aineistossa esiintyvät henkilöt ja heidän esiintymänsä tekstissä.

Vaikka työkalu onkin lähtökohtaisesti toteutettu vain yhtä aineistoa varten, voidaan sillä helposti nähdä olevan myös muita käyttötarkoituksia. Tämä pätee sekä annotointi- että anonymisointitoiminnallisuuteen sekä yhdessä että erikseen. Eräs kiinnostava anonymisoinnin sovellusalue olisi Turussa kerätty teho-osaston potilaskertomusaineisto [Haverinen, Ginter, Laippala ja Salakoski 2009]. Aineisto on anonymisoitu manuaalisesti [Ginter et al. 2010], joten anonymisointia suorittavalle ohjelmistolle voidaan olettaa olevan käyttöä myös laajemmin.

6.1.1 Vertailu vastaaviin työkaluihin

Lähimmäksi Anonymisoivan annotaattorin toiminnallisuutta osuu **Anonymizer for Finnish documents** (AFFD) [AFFD 2011]. Se poistaa tai korvaa dokumenteista henkilö- ja yhteystietoja. Alunperin AFFD:tä tarkastellessani luovuin sen käytöstä ilmoitettujen ominaisuuksien perusteella, koska se vaikutti vaativan liian paljon käyttäjän työtä anonymisoinnissa. Uudemmassa tätä työtä varten tekemässäni tarkastelussa

kuitenkin huomasin, että se osaa taivuttaa nimiä korvaamista varten, mikäli sille vain annetaan nämä nimet. AA toimii kuitenkin AFFD:tä automaattisemmin suorittaen nimien tunnistamisen ja korvaamisen parhaassa tapauksessa täysin ilman käyttäjän väliintuloa. AFFD taas vaatii käyttäjää antamaan korvaavan nimen.

Anonymisoivan annotaattorin ja AFFD:n toiminnallisuus on osittain päällekkäistä Karkeasti voidaan todeta AFFD:n tekevän melko lailla samat asiat kuin AA:n anonymisointiosan. Anonymisoiva annotaattori tekee anonymisoinnin automaattisemmin, mutta ei sisällä varsinaista käyttöliittymää. Mikäli prosessiin tarvitsee puuttua, tapahtuu se muokkaamalla AA:n syötetiedostoja. AFFD taas vaatii enemmän käyttäjän suoraa puuttumista työprosessiin, mutta tarjoaa sitä varten WWW-käyttöliittymän. Kaiken kaikkiaan AFFD on viimeistellympi ja jo julkaistu sovellus. AA taas kaipaa vielä paljon viimeistelyä, ennen kuin se on käyttökelpoinen samalle käyttäjäryhmälle.

6.1.2 Käytetyt kielivarat

Ohjelmisto käyttää sanastokantanaan Joukahainen-sanastotietokantaa [Pitkänen 2007]. Kyseinen sanastotietokanta sisältää suomalaisia sanoja ja niihin liittyvää metatietoa esimerkiksi sanan taipumiseen ja sanaluokkaan liittyen. Joukahaisessa osa sanoista on merkitty eris- tai sukunimiksi ja nämä sanalistat toimivat pohjana sovelluksen käyttämille sanalistoille. Listojen laajentamiseen on käytetty myös suomenkielistä Wikipediaa ja sieltä löytyviä etu- ja sukunimilistoja. Lisäksi nimilistoja on laajennettu aineiston nimien perusteella sekä muista sekalaisista lähteistä.

Muita mahdollisia kielivaroja

Anonymisoivan annotaattorin nykyinen versio luokittelee nimet kolmeen eri luokkaan miesten ja naisten nimiin sekä sukunimiin. Kunkin nimen oletetaan kuuluvan vain yhteen luokkaan. Nimeen liitetyllä sukupuolella on merkitystä silloin, kun ohjelma valitsee uutta nimeä anonymisointia varten, joten miehen nimi korvataan miehen nimellä ja niin edelleen. Käytännössä tilanne ei kuitenkaan aina ole näin yksinkertainen. Esimerkkinä tästä käy taulukko 6.1, jossa on Väestörekisterikeskuksen nimipalvelun antamaa tilastotietoa nimen *Kaino* yleisyydestä [VRK 2012]. Kuten taulukosta huomataan, tilastojen valossa nimen *Kaino* antaminen ei ole selkeästi sukupuolittunutta. Mikäli ohjelmalla olisi vastaavat tiedot käytettävissään, voisi nimien luokittelu perustua tilastoihin, eikä pelkästään nimilistan kerääjän mielivaltaiseen päätökseen. Koska tilastoissa on mukana myös ajallista informaatiota, voidaan sen avulla parantaa arvausta nimen kantajan sukupuolesta, mikäli tiedetään henkilön summittainen ikä.

Väestörekisterikeskuksen nimipalvelua käyttämällä olisi mahdollista saada laajempaa tietoa nimistä ja niiden yleisyydestä. Tätä tietoa voitaisiin käyttää helpottamaan monitulkintaisten nimien kuten *Valo* tunnistamista. Väestörekisterikeskuksen kirjan-

pidon mukaan *Valo*-nimisiä henkilöitä on ollut hieman yli kuusisataa. Anonymisointia varten olisi näin ollen hyödyllistä tietää kaikki Väestörekisterikeskuksen kirjanpidossa olevat etu- ja sukunimet.

Taulukko 6.1: Väestörekisterikeskuksen nimipalvelun tuloste Kaino-nimen yleisyydestä

Syntymävuodet	Miehiä	Naisia	Yhteensä
-1899	28	51	79
1900-19	1226	908	2134
1920-39	1428	1395	2823
1940-59	267	246	513
1960-79	25	17	42
1980-99	14	8	22
2000-09	17	15	32
2010-12	6	6	12
Yhteensä	3011	2646	5657

6.2 RDF-NLP

Aineiston lopullinen tallennusmuoto perustuu RDF-kieleen. Tallennusmuoto ja RDF-kieli on kuvattu tarkemmin osiossa 4. Seuraavassa esitellään lyhyesti RDF-muotoon tallennetun tekstiaineiston käsittelyyn tarkoitettua RDF-NLP -ohjelmistoa.

Pääosa jäsennetyn aineiston käsittelystä tapahtuu RDF-muotoon muuntamisen jälkeen. Tähän tarkoitukseen on toteutettu RDF-NLP -ohjelmisto, joka on saanut nimensä sanoista RDF (Resource Description Framework) ja NLP (natural language processing). Ohjelmiston toiminnallisuus koostuu RDF-muotoisen tekstin käsittelyä varten toteutetuista perusprimitiiveistä, joilla tekstiä voidaan käsitellä sana ja lause kerrallaan. RDF-NLP ohjelmisto on toteutettu Prolog-kielellä ja sen avulla esimerkiksi tapauksessa esiintyvien henkilöiden roolien päättely voidaan muotoilla deklaratiivisesti graafin täsmäys -tehtävänä kohdan 5.2.1 tapaan. Itsessään RDF-NLP sisältää vain suppean toiminnallisuuden aineiston käsittely varten. Se ei esimerkiksi koosta lauseista laajempia kokonaisuuksia tai edes tallenna niiden keskinäistä järjestystä. Tämä ei kuitenkaan ole ongelma, koska RDF-muotoon tallennettuun aineistoon on helppo lisätä uusia predikaatteja. Uusien predikaattien lisääminen RDF-solmujen välille ei myöskään haittaa RDF-NLP:n toimintaa, mikäli lisättävät predikaatit eivät ole nimiltään päällekkäisiä sen käyttämien kanssa. RDF-NLP:n avulla voidaan siis yhtä hyvin käsitellä pelkistä sana- ja lauseolioista koostuvaa aineistoa kuin saman aineiston annotoitua ja henkilöesiintymäinformaatiolla annotoitua muotoa. Tätä ominaisuutta on hyödynnetty, kun RDF-NLP:n päälle on toteutettu aineiston käsittelyn kirjauksittain mahdollistavat pri-

mitiivit. Samalla tavalla kerrostamalla on lopulta toteutettu myös aineiston henkilöiden roolien päättely sekä valtaosa kohdassa 5 esitellystä toiminnallisuudesta.

6.3 Käytettyjen työkalujen esittely

Työssä toteutetun toiminnallisuuden lisäksi työssä on käytetty mahdollisuuksien mukaan jo olemassa olevia ohjelmistoja. Alkuperäinen aineisto on muutettu alkuperäisestä PDF-muodosta tekstiksi **pdftotext** työkalulla. Esikäsittelyvaiheessa käytettyjä työkaluja on esitelty tarkemmin jo aiemmin kohdassa 3.1, mutta todettakoon, että esikäsittelyssä hyödynnetään SWERG+ -taivutusmuotogeneraattoria [Kettunen ja Arvola 2012] sekä Suomi-malaga -työkalua [Väisänen ja Pitkänen 2006]. NLTK-kirjastoa [Bird, Loper ja Klein 2009] käytetään lisäksi anonymisointivaiheessa aineiston hallintaan.

Ennen jäsennystä aineisto jaetaan lauseiksi NLTK-kirjaston Punkt-lauseidenerottajaa [Kiss ja Strunk 2006] käyttäen. Tämän jälkeen lauseet jäsennetään CSC:n palvelimella käyttäen fi-fdg -jäsennintä [Tapanainen ja Järvinen 1997]. Etäjäsennysvaiheessa hyödynnetään GNU Parallell -työkalua [Tange 2011], jota käyttäen suoritetaan jäsentimen käyttäminen verkon yli. Jäsentimen XML-muotoinen jäsennystulos muutetaan RDF-muotoon käyttäen apuna RDFLib-kirjastoa [RDFLib 2002].

Anonymisoiva annotaattori on toteutettu Python- ja Prolog-kielillä siten, että anonymisointiosassa on käytetty Pythonia ja annotoinnissa käytetään Prologia sekä edellä esiteltyä RDF-NLP -kirjastoa laajennuksineen. RDF-NLP:n toiminta nojaa täysin SWI-Prologin [Wielemaker et al. 2012] RDF-kirjastoon. Prologin unifikaatio-mekanismia hyödynnetään varsin laajasti työssä ja myös osittaisten nimen esiintymien yhdistäminen aineiston henkilöihin on toteutettu sitä käyttäen.

Aineistosta tehdyt aikajanat on tuotettu käyttäen R-kieltä [R Core Team 2012]. Graafivisualisaatiot taas on tuotettu Gephi-graafieditorilla [Bastian, Heymann ja Jacomy 2009], jota voidaan käyttää myös graafien suodattamiseen. Aineiston muuntaminen ja vienti Gephin tukemaan GEXF (Graph Exchange XML Format) [GEXF-WG 2009] -tiedostomuotoon on toteutettu Prologilla.

6.4 Muut mahdolliset työkalut

Työssä käytettyjen työkalujen lisäksi luonnollisen kielen käsittelyyn löytyy laaja joukko muita ohjelmistoja. Tyypillistä niille on suomen kielen tuen puuttuminen sekä koneoppimisen avulla tuotettujen mallien käyttäminen. Valmiita malleja ei kuitenkaan suomen kieltä varten ole saatavilla. Näin ollen nämä ohjelmistot ovat rajautuneet työssä käytettyjen työkalujen ulkopuolelle. Esimerkkeinä tällaisista ohjelmistoista ovat OpenNLP [ASF 2010], GATE [Cunningham et al. 2013] ja UIMA [Ferrucci ja Lally 2004]. Mallipainotteisuutensa lisäksi ne kaikki ovat varsin laajoja ja monipuolisia sovelluskehyksiä luonnollisen kielen käsittelyyn.

Vapaasti saatavilla olevia laajoja suomenkielisiä koneoppimista varten annotoituja aineistoja on tiettävästi olemassa vain muutamia. FinnTreeBank [Voutilainen, Purtonen ja Muhonen 2012] -puupankin kolmas versio sisältää joukon automaattisesti dependenssijäsennettyjä lauseita, joita voitaisiin käyttää aiemmin mainittujen työkalujen vaatimien kielellisten mallien oppimiseen. Toinen vastaava, mutta käsin annotoitu puupankki on Turku Dependency Treebank [Haverinen, Ginter, Laippala, Kohonen et al. 2011]. Ei ole kuitenkaan täysin selvää, kuinka hyvin puupankinkin pohjalta opittu malli soveltuisi käytettyyn lastensuojelun asiakaskertomusaineistoon ja sen kielellisiin erityispiirteisiin. Myöskään työn aineiston pohjalta ei päädytty toteuttamaan koneoppismallia, koska riittävän opetusaineiston annotoiminen käsin todettiin liian työlääksi sekä laajempaa kielitieteellistä asiantuntemusta vaativaksi.

7 Yhteenveto

Työssä on tarkasteltu lastensuojelun asiakaskertomusten rakenteistamista, niiden sisältämän informaation harmonisointia, anonymisointia, visualisointia sekä tiedon eristämistä. Päätuotteena on joukko eri työvaiheita toteuttavia ohjelmistoja sekä käyttökokemuksia niiden soveltamisesta lastensuojelun asiakaskertomusaineistoon.

Haasteena työssä on ollut suomen kieltä tukevien resurssien ja ohjelmistojen huono saatavuus. Toisaalta taas luonnollisen kielen käsittelyyn löytyy paljon erilaisia ohjelmistoja ja lähestymistapoja, joiden määrä tuo mukanaan valinnanvaikeuden, kun etsitään kuhunkin tarkoitukseen parhaiten sopivan työkalua. Lisäksi oman haasteensa työhön on tuonut sen monitieteisyys sosiaalityön, kieliteknologian, kielitieteen ja tietojenkäsittelyn leikkauspisteessä.

Työn aikana on toteutettu useita eri työvaiheita automatisoivia ohjelmistoja. Anonymisointi ja sitä edeltävä nimien tunnistaminen toimii aineistolla hyvin, kunhan tuntemattomat nimet ensin lisätään järjestelmään. Myöskään ulkomaalaiset nimet eivät tuota lisäämisen jälkeen ongelmia. Korvaavien nimien taivuttaminen ja valitseminen sujuu pääosin hyvin, pois lukien tietyt nimet, joiden oikea taivuttaminen on osoittautunut SWERG+ -taivutusmuotogeneraattorille haastavaksi. Osa nimien taivutuksista onkin nimienkorvaussääntöjä varten korjattu käsin. Väärin taivutettuja ja valittuja korvaavia nimiä oli ennen korjausta alle kymmenesosa kaikista taivutusmuodoista. Nykyisellään SWERG+:n tuottamat taivutusmuodot nimille on tallennettu tietokantaan. Ongelman vaikutusta voitaisiin vähentää korjaamalla väärin taivutetut taivutusmuodot kannassa ja käyttämällä näitä korjattuja taivutusmuotoja, mikäli sellaisia on, alkuperäisten sijaan.

Kirjausrajojen erottelu on toteutettu sääntöpohjaisesti, vaikka myös koneoppimisvaihtoehtoa harkittiin. Sääntöpohjainen lähestymistapa on kuitenkin aineiston säännönmukaisuuden vuoksi erittäin tarkka ja kirjausrajoja on lisätty käsin 167 kappaletta. Kaikkiaan kirjauksia on 2220 kappaletta.

Kirjausten pilkkominen lauseisiin ja lauseiden dependenssijäsentäminen toteutettiin valmiilla työkaluilla. Lauseiden erottaminen toimi valmista mallia käyttäen hyvin, eikä aineiston pohjalta tästä syystä opetettu uutta lauseidenerotusmallia. Dependenssijäsentäminen taas toimi hieman huonommin, oletettavasti johtuen aineiston kielellisistä erityispiirteistä. Jäsentimen tuntemaan sanastoon voi lisätä uusia sanoja, mutta näin ei työssä tehty. Sosiaalityöntekijöiden ammattisanaston lisääminen saattaisi kuitenkin jatkossa parantaa jäsennystulosta.

Annotointi ja nimien esiintymien liittäminen tapauksen henkilöihin onnistui pääosin hyvin. Ongelmien lähteenä olivat lähinnä aineistossa tapahtuvat nimien muutokset, esimerkkinä sijaisperheeseen sijoitetun lapsen sukunimen vaihtuminen samaksi kuin sijaisperheellä. Myös muita nimien muutoksia esiintyi, mutta niitä ei oltu eksplisiittisesti kirjattu asiakaskertomuksiin. Toinen nimien yhdistämistä vaikeuttava seikka oli

lempinimien tai lyhennenimien käyttö. Esimerkkinä lempi- ja lyhennenimien käytöstä voidaan ottaa nimi *Jarkko-Ilari*, joka saattaa kirjauksissa esiintyä myös muodossa *J-I*, *Jarkko* tai *Jake*. Nimien yhdistämisen perustuessa nimien päällekkäisyyteen ei järjestelmä nykyisellään osaa päätellä nimen *Jarkko* liittyvän samaan henkilöön kuin nimen *Jarkko-Ilari Martikainen*. Ongelma on työssä korjattu suorittamalla anonymisointi iteratiivisesti. Aineisto on ensin anonymisoitu, jonka jälkeen anonymisointitulos on tarkastettu. Löydetyt anonymisoimattomat nimet on lisätty anonymisoijan tuntemien nimien joukkoon, jonka jälkeen anonymisointi on suoritettu uudestaan. Samalla samaa henkilöä tarkoittavat eri nimet on määritelty synonyyminimiksi, jolloin aineiston lopullisessa anonymisoidussa muodossa kullakin henkilöllä on yksikäsitteinen nimi. Esimerkkinä voidaan ottaa anonymisoimattoman aineiston synonyyminimet *Jarkko* ja *Jake*, jotka kuvautuvat anonymisoidussa aineistossa nimelle *Aki*.

Taulukossa 7.1 on esitetty kolmen tapauksen anonymisoinnissa käytettyjen korvaussääntöjen lukumäärät sekä niihin tehtyjen korjausten lukumäärät. Kolmannessa sarakkeessa on laskettu lisättyjen ja korjattujen sääntöjen osuus kaikista säännöistä. Tehdyt lisäykset ja korjaukset koostuvat käsin korjatuista nimien korvaussäännöistä ja lisätyistä korvaussäännöistä. Korvaussääntöjä on lisätty niille nimille, joiden taivutusmuotoja ei ole tunnistettu aineistosta. Prosenttiosuuden voidaan ajatella kuvaavan nimien automaattisen anonymisoinnin onnistumista ja anonymisoinnissa tarvittavan manuaalisen työn osuutta.

Taulukko 7.1: Anonymisoinnissa käytettyjen korvaussääntöjen määriä ja korjattujen sääntöjen osuuksia.

Korvaussääntöjen lukumäärä	Muutettuja tai lisättyjä korvaussääntöjä	Korjausten osuus
98	23	23%
415	68	16%
235	60	25%

Tapausten henkilöiden roolien päättely toimii nykyisellään hyvin, mutta pääteltyjen roolien joukko vaatii vielä osin karsintaa. Tämä koskee etenkin tapauksia, joissa useammalle henkilölle on tunnistettu esimerkiksi rooli “äiti”. Tässä esimerkkitapauksessa olisi kiinnostavaa kyetä erottamaan lapsen äiti sijaisäidistä. Laajempi ja tarkempi tiedon eristäminen aineistosta vaatisi monipuolisempia lähestymistapoja jäsennyspuun hyödyntämisen lisäksi. Tuloksia voitaisiin kohentaa myös parantamalla tässä työssä käytettyjä, varsin yksinkertaisia tapoja hyödyntää jäsennyspuita. Nykyinen menetelmä löytää eräässä aineiston tapauksessa useimmin mainittujen henkilöiden rooleista noin puolet. Tapauksessa on 25 henkilöä, jotka mainitaan kymmenen kertaa tai useam-

min ja näistä 12 tunnistetaan rooli. Puuttuvat roolit johtuvat siitä ettei henkilöillä ole kirjattu roolia tai nykyinen menetelmä ei tunnista sitä. Ongelmia aiheuttavat esimerkiksi lauseet jotka ovat muotoa “Paikalla sosiaalityöntekijät Makkonen ja Rantala”. Näistä lauseista roolin päättely onnistuu nykyisellään vain ensimmäisenä mainitulle henkilölle.

Työn tuloksia on esitelty aineiston luovuttaneen kaupungin sosiaalityöntekijöille 22.1.2013. Esittelyn pääpainona olivat visualisaatiot, koska niiden arveltiin olevan kohdeyleisön kannalta kiinnostavin osa työstä. Niistä saatiinkin hyvää palautetta, etenkin henkilöaikajanan osalta. Henkilöaikajanan todettiin tuovan selkeästi esille lastensuojelun laajan sosiaalisen ulottuvuuden, koska aikajana tuo kerralla näkyviin sen, ettei lastensuojelutyö ole rajautunut pelkästään lapseen, tämän perheeseen ja sosiaalityöntekijään. Aikajanalle löydettiin heti myös käyttötapaus ja eräs sosiaalityöntekijä kertoi työnsä nopeutuvan ja helpottuvan, mikäli hänellä olisi käytössään tässä työssä toteutettua toiminnallisuutta. Hyötyä saavutettaisiin etenkin erilaisten kirjallisten yhteenvetojen koostamisessa, joita tehdään esimerkiksi perheen muuttaessa toiselle paikkakunnalle ja samalla siirtyessä uuden kunnan sosiaalityöntekijöiden vastuulle.

8 Johtopäätökset

Sosiaalityöntekijöiden työotetta on kuvattu holistiseksi, jolloin heidän tavoitteenaan on ymmärtää asiakkaat kokonaisuutena ja osana sosiaalista kontekstia [Huuskonen ja Vakkari 2011]. Tässä työssä on tuotettu toiminnallisuutta tukemaan sosiaalityöntekijöitä työssään ja auttamaan heitä saamaan nopeasti kuva asiakkuuksien historiasta ja sosiaalisesta kontekstista. Jo työn varsin yksinkertaisin menetelmin ja visualisoinnein on osoitettu, että jo olemassa olevia kirjauksia voidaan automaattisesti muuntaa havainnollisempaan muotoon. Tällaisenaan työ antaa hyvän lähtökohdan jatkotutkimukselle ja kehitystyölle. Erityisen kiinnostavaa olisi tehdä lähempää yhteistyötä sosiaalityöntekijöiden kanssa, jotta aineiston sisällön analysointia ja siitä tehtäviä visualisointeja voitaisiin toteuttaa varsinaisia loppukäyttäjiä paremmin kuunnellen.

Lastensuojelun asiakaskertomukset tarjoavat kiinnostavan aineiston myös muille, eri näkökulmista aihetta lähestyville, tarkasteluille. Aineistoa voitaisiin lähestyä puhtaan kielellisestä näkökulmasta ja tarkastella, millaista kieltä kirjauksissa käytetään ja kenestä niissä kerrotaan. Lastensuojelussa lähtökohtana on lapsi ja lapsen etu [Lastensuojelulaki 2007, § 4], mutta osassa aineiston kirjauksia yllättävän suuren roolin saavat kuitenkin aikuiset. Edellä esitetty on vain yksi esimerkki niistä monista tutkimusnäkökulmista, joista aineistoa voidaan lähestyä, työssä kuvatun aineiston automaattisen visualisoinnin ja tiedon eristämisen lisäksi.

Kirjallisuus

[AFFD 2011] AFFD team. *affd - Anonymizer for finnish documents*. 2011. URL: http://code.google.com/p/affd/ (viitattu 29. 10. 2012) (ks. s. 54).

[ASF 2010] Apache Software Foundation. *OpenNLP*. 2010. URL: http://opennlp.apache.org (viitattu 29. 05. 2013) (ks. s. 57).

[Bastian, Heymann ja Jacomy 2009] Mathieu Bastian, Sebastien Heymann ja Mathieu Jacomy. *Gephi: An Open Source Software for Exploring and Manipulating Networks*. 2009. URL: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154 (ks. s. 40, 57).

[Berners-Lee, Fielding ja Masinter 2005] T. Berners-Lee, R. Fielding ja L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986 (INTERNET STANDARD). Updated by RFC 6874. Internet Engineering Task Force, 2005. URL: http://www.ietf.org/rfc/rfc3986.txt (ks. s. 27, 28).

[Beutel 1995] Björn Beutel. *Malaga 7.12. Malaga kielen ja ohjelmiston dokumentaatio*. 1995. URL: http://home.arcor.de/bjoern-beutel/malaga/malaga.html (ks. s. 12).

[Bird, Loper ja Klein 2009] Steven Bird, Edward Loper ja Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009 (ks. s. 18, 57).

[Brickley ja L. Miller 2010] Dan Brickley ja Libby Miller. *FOAF Vocabulary Specification 0.97*. Namespace document. 2010. URL: http://xmlns.com/foaf/spec/20100101.html (ks. s. 28).

[Carroll ja Klyne 2004] Jeremy J. Carroll ja Graham Klyne. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation. W3C, 2004. URL: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ (ks. s. 26, 27).

[Connexor 2006] Connexor Oy. *Machinese Language Model Manual*. 2006 (ks. s. 22).

[Cunningham et al. 2013] Hamish Cunningham, Valentin Tablan, Angus Roberts ja Bontcheva Kalina. "Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics". *PLoS Compututational Biology* 9.2 (2013) (ks. s. 57).

[Davis-Mendelow 1998] Steven Davis-Mendelow. "Bridging the gap: Exploring the information needs and information use of front line child protection intake workers." väitöskirja. University Of Toronto, 1998 (ks. s. 1).

[Elmasri ja Navathe 2004] Ramez Elmasri ja Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Boston, MA, USA: Addison-Wesley, 2004. ISBN: 0-321-20448-4 (ks. s. 8, 14, 15).

[Etzioni et al. 2008] Oren Etzioni, Michele Banko, Stephen Soderland ja Daniel S. Weld. "Open information extraction from the web". *Communications of the ACM* 51 (12 joulukuuta 2008), s. 68–74. ISSN: 0001-0782. URL: `http://doi.acm.org/10.1145/1409360.1409378` (ks. s. 1).

[Ferrucci ja Lally 2004] David Ferrucci ja Adam Lally. "UIMA: an architectural approach to unstructured information processing in the corporate research environment". *Natural Language Engineering* 10.3-4 (2004), s. 327–348 (ks. s. 57).

[Finin ja Palmer 1983] Timothy W. Finin ja Martha Stone Palmer. "Parsing with logical variables". Teoksessa: *Proceedings of the first conference on Applied natural language processing*. Toim. Iris Kameny. Santa Monica, California: Association for Computational Linguistics, 1983, s. 62–68. URL: `http://www.aclweb.org/anthology-new/A/A83/` (ks. s. 24).

[Lastensuojelulaki 2007] Finlex, toim. *Lastensuojelulaki (417/2007)*. 2007. URL: `http://www.finlex.fi/fi/laki/ajantasa/2007/20070417` (ks. s. 1, 3, 62).

[GEXF-WG 2009] GEXF Working Group. *GEXF (Graph Exchange XML Format)*. 2009. URL: `http://gexf.net` (viitattu 29. 05. 2013) (ks. s. 57).

[Ginter et al. 2010] Filip Ginter, Katri Haverinen, Veronika Laippala ja Timo Viljanen. *Turku Clinical TreeBank and PropBank*. 2010. URL: `http://bionlp.utu.fi/clinicalcorpus.html` (viitattu 27. 05. 2013) (ks. s. 2, 54).

[Hausser 1992] Roland Hausser. "Complexity in Left-Associative Grammar". *Theoretical Computer Science* 106.2 (1992), s. 283–308 (ks. s. 12).

[Haverinen, Ginter, Laippala, Kohonen et al. 2011] Katri Haverinen, Filip Ginter, Veronika Laippala, Samuel Kohonen, Timo Viljanen, Jenna Nyblom ja Salakoski Tapio. "A Dependency-based Analysis of Treebank Annotation Errors". Teoksessa: *Proceedings of International Conference on Dependency Linguistics*. Toim. Kim Gerdes, Eva Hajicova ja Leo Wanner. 2011 (ks. s. 58).

[Haverinen, Ginter, Laippala ja Salakoski 2009] Katri Haverinen, Filip Ginter, Veronika Laippala ja Tapio Salakoski. "Parsing Clinical Finnish: Experiments with Rule-Based and Statistical Dependency Parsers". Teoksessa: *Proceedings of NODALIDA 2009*. Toim. Kristiina Jokinen ja Eckhard Bick. NEALT, 2009, s. 65–72 (ks. s. 7, 54).

[Hiissa et al. 2006] Marketta Hiissa, Tapio Pahikkala, Hanna Suominen, Tuija Lehtikunnas, Barbro Back, Eija Helena Karsten, Sanna Salanterä ja Tapio Salakoski. "Towards automated classification of intensive care nursing narratives". Teoksessa: *Ubiquity: Technologies for Better Health in Aging Societies. Proceedings of MIE 2006, the 20th International Congress of the European Federation of Medical Informatics*. Toim. A Hasman, R Haux, J Van Der Lei, De Clercq E ja FH Roger France. Vol. 124. IOS Press, 2006, s. 789–794 (ks. s. 2).

[Huuskonen ja Vakkari 2010] Saila Huuskonen ja Pertti Vakkari. "Client information system as an everyday information tool in child protection work". Teoksessa: *Third Symposium on Information Interaction in Context*. Toim. Nicholas J. Belkin ja Diane Kelly. 2010 (ks. s. 1, 2).

[Huuskonen ja Vakkari 2011] Saila Huuskonen ja Pertti Vakkari. "Client's Temporal Trajectory in Child Protection: Piecing Information Together in a Client Information System". Teoksessa: *Human-Computer Interaction - INTERACT 2011*. Toim. Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque ja Marco Winckler. Vol. 6949. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, s. 152–169 (ks. s. 1, 62).

[Kääriäinen, Leinonen ja Metsäranta 2006] Aino Kääriäinen, Ansa Leinonen ja Hannele Metsäranta. *Lastensuojelutyön dokumentointi*. Helsinki: Yliopistopaino, 2006 (ks. s. 43).

[Karlsson 2008] Fred Karlsson. *Yleinen kielitiede*. Helsinki: Gaudeamus Helsinki University Press, 2008. ISBN: 978-952-495-071-8 (ks. s. 20, 23, 49).

[Kettunen ja Arvola 2012] Kimmo Kettunen ja Paavo Arvola. "Generating Variant Keyword Forms for a Morphologically Complex Language Leads to Successful Information Retrieval with Finnish. 5th International Retrieval Facility Conference, IRFC 2012, Vienna, Austria, July 2-3, 2012 Proceedings". Teoksessa: *Multidisciplinary Information Retrieval*. Toim. Michail Salampasis ja Birger Larsen. Lecture Notes in Computer Science. Springer, 2012, s. 113–126. ISBN: 978-3-642-31273-1 (ks. s. 12, 13, 57).

[Kiss ja Strunk 2006] Tibor Kiss ja Jan Strunk. "Unsupervised Multilingual Sentence Boundary Detection". *Computational Linguistics* 32.4 (2006), s. 485–525 (ks. s. 18, 57).

[Krötzsch et al. 2007] Markus Krötzsch, Denny Vrandecic, Max Völkel, Heiko Haller ja Rudi Studer. "Semantic Wikipedia". *Journal of Web Semantics* 5.4 (2007), 251–261 (ks. s. 1, 50).

[Kuurne 2009] Salla Kuurne. *Sähköinen, rakenteinen kirjaaminen hoitotyössä – miten hoitotyö tulee näkyväksi. Esimerkki erikoissairaanhoidosta: psykiatrian, sisätautien vuodeosasto kirurginen vuodeosasto ja poliklinikka*. FCG Finnish Consulting Group Oy. 3. syyskuuta 2009. URL: {http://www.lshp.fi/download.aspx?ID=2149&GUID=%7BCC6868BF-AAE9-4509-ACFF-0A0987A9D679%7D} (viitattu 29. 05. 2013) (ks. s. 53).

[Laippala et al. 2009] Veronika Laippala, Filip Ginter, Sampo Pyysalo ja Tapio Salakoski. "Towards Automated Processing of Clinical Finnish: A Sublanguage Analysis and a Rule-Based Parser". *International Journal of Medical Informatics, Special Issue on Mining of Clinical and Biomedical Text and Data* 78.12 (2009), e7–e12 (ks. s. 2).

[E. Miller ja Manola 2004] Eric Miller ja Frank Manola. *RDF Primer*. W3C Recommendation. W3C, 2004. URL: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ (ks. s. 26, 27).

[Nguyen 2007] Loc H. Nguyen. "Child welfare informatics: A new definition for an established practice". *Social Work* 52.4 (2007), s. 361–363 (ks. s. 2).

[Nothman et al. 2012] Joel Nothman, Edward Loper, Steven Bird ja Willy. *Natural Language Toolkit: Punkt sentence tokenizer*. 2012. URL: http://nltk.org/_modules/nltk/tokenize/punkt.html (ks. s. 18).

[Pitkänen 2007] Harri Pitkänen, toim. *Joukahainen - WWW-pohjainen suomen kielen sanastotietokanta*. 2007. URL: http://joukahainen.puimula.org/ (ks. s. 55).

[Poesio, Ponzetto ja Versley 2011] Massimo Poesio, Simone Ponzetto ja Yannick Versley. "Computational models of anaphora resolution: A survey". *Linguistic Issues in Language Technology* (2011) (ks. s. 47).

[R Core Team 2012] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. 2012. ISBN: 3-900051-07-0. URL: http://www.R-project.org/ (viitattu 28. 05. 2013) (ks. s. 57).

[RDFLib 2002] RDFLib-kehittäjät. *RDFLib*. 2002. URL: https://github.com/RDFLib/rdflib (viitattu 28. 05. 2013) (ks. s. 57).

[Strunk 2009] Jan Strunk. *Pretrained models for PUNKT sentence boundary detector*. Sprachwissenschaftliches Institut, Ruhr-Universität Bochum. 24. elokuuta 2009. URL: https://groups.google.com/forum/#!msg/nltk-dev/y2zYJSOdevQ/an2EKpBOBD0J (ks. s. 18).

[FTC 2000]. *Suomen kielen tekstikokoelma*. 2000. URL: http://www.csc.fi/english/research/software/ftc (ks. s. 18).

[Suominen 2007] Hanna Suominen. ”Kieliteknologian menetelmien soveltaminen potilasdokumentaation hyödyntämiseen (Applying human language technology to utilization of patient documentation)”. Teoksessa: *Tietojenkäsittelytieteen päivät 2007*. Toim. M Koskinen ja E Jauhiainen. Jyväskylän yliopistopaino, Jyväskylä, 2007, s. 46–50 (ks. s. 2).

[Tange 2011] Ole Tange. ”GNU Parallel - The Command-Line Power Tool”. *;login: The USENIX Magazine* 36.1 (2011), s. 42–47. URL: `http://www.gnu.org/s/parallel` (ks. s. 57).

[Tapanainen ja Järvinen 1997] Pasi Tapanainen ja Timo Järvinen. ”A non-projective dependency parser”. Teoksessa: *Proceedings of the Fifth Conference on Applied Natural Language Processing*. Toim. P. Jakobs. Association for Computational Linguistics, 1997, s. 64–71 (ks. s. 1, 20, 57).

[Tarviainen 1977] Kalevi Tarviainen. *Dependenssikielioppi*. Helsinki: Oy Gaudeamus Ab, 1977. ISBN: 951-662-197-X (ks. s. 1, 9, 20).

[Taskinen 2010] Sirpa Taskinen. *Lastensuojelulain soveltaminen*. Helsinki: WSOYpro, 2010. ISBN: 978-951-0-36845-9 (ks. s. 3).

[THL 2007] Terveyden ja hyvinvoinnin laitos, toim. *Lastensuojelun käsikirja*. 2007. URL: `http://www.sosiaaliportti.fi/tietoa_palvelusta` (viitattu 22. 05. 2013) (ks. s. 38).

[VRK 2012] Väestörekisterikeskus. *Väestörekisterikeskuksen nimipalvelu*. 2012. URL: `http://verkkopalvelu.vrk.fi/Nimipalvelu/default.asp?L=1` (ks. s. 55).

[Väisänen ja Pitkänen 2006] Hannu Väisänen ja Harri Pitkänen. *Suomi-malaga - suomen kielen muoto-opin kuvaus*. 2006. URL: `https://github.com/voikko/corevoikko/tree/master/suomimalaga` (ks. s. 12, 57).

[Voutilainen, Purtonen ja Muhonen 2012] Atro Voutilainen, Tanja Purtonen ja Kristiina Muhonen, toim. *FinnTreeBank - A Treebank for Finnish*. 2012. URL: `http://www.ling.helsinki.fi/kieliteknologia/tutkimus/treebank/index.shtml` (viitattu 29. 05. 2013) (ks. s. 58).

[Weal et al. 2007] Mark J. Weal, Harith Alani, Sanghee Kim, Paul H. Lewis, David E. Millard, Patrick A. S. Sinclair, David C. De Roure ja Nigel R. Shadbolt. ”Ontologies as facilitators for repurposing web documents”. *International Journal of Human-Computer Studies* 65 (6 2007), s. 537–562. ISSN: 1071-5819 (ks. s. 25).

[Wielemaker et al. 2012] Jan Wielemaker, Tom Schrijvers, Markus Triska ja Torbjörn Lager. ”SWI-Prolog”. *Theory and Practice of Logic Programming* 12.1-2 (2012), s. 67–96 (ks. s. 57).

[Wu ja Weld 2010] Fei Wu ja Daniel S. Weld. ”Open information extraction using Wikipedia”. Teoksessa: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL ’10. Uppsala, Sweden: Association for Computational Linguistics, 2010, s. 118–127 (ks. s. 1).